Identification of dangerous contingencies for large scale power system security assessment

PhD dissertation by
Florence Fonteneau-Belmudes

Department of Electrical Engineering and Computer Science,
University of Liège, Belgium
2011

Foreword

When I was studying electrical engineering and computer science at the French Grande Ecole *SUPELEC* (*Ecole Supérieure d'Electricité*), I had the opportunity to work with Prof. Damien Ernst, who was supervising my final internship. This internship was carried out in the Island Energy Systems division of EDF in Corsica. I mainly worked there on the prediction, analysis and mitigation of $N-1$ contingencies on the 90 kV transmission network.

After graduating from *SUPELEC*, I was given the chance to begin a PhD in the Electrical Engineering and Computer Science department of the *University of Liège* (Belgium) under the supervision of Prof. Damien Ernst and Prof. Louis Wehenkel, on the problem of identifying dangerous contingencies for large scale power system security assessment with bounded computational resources.

Acknowledgements

First and foremost, I would like to express my deepest gratitude and appreciation to Prof. Louis Wehenkel for offering me the opportunity to discover the world of research. His remarkable talent and research experience have been very valuable at every stage of this work.

I would like to extend my deepest thanks to Prof. Damien Ernst, for his up to the point suggestions and advice regarding every research contribution reported in this dissertation. He proved to be a great collaborator on many scientific and human aspects. His enthusiasm, creativity and constant support have been of great importance all along my PhD.

My deepest gratitude also goes to Christophe Druet, and to the whole ELIA company, for providing me with data and simulation tools to apply the approach developed in this thesis to the Belgian transmission network as well as valuable feedback on the developed framework.

Very special thanks go to Prof. Mania Pavella for her kindness and the example she is for every woman in the field of power system research.

I also address my warmest thanks to all the SYSTMOD research unit and the Department of Electrical Engineering and Computer Science, where I found a friendly and stimulating research environment. Many thanks to the academic staff and especially Dr Florin Capitanescu, Dr Mevludin Glavic and Prof. Thierry Van Custem. A special acknowledgement to my office neighbors, Bertrand Cornélusse, Renaud Detry, Samuel Hiard and Da Wang. Many additional thanks to Petros Aristidou, Julien Becker, Vincent Botta, Anne Collard, Boris Defourny, Guillaume Drion, Davide Fabozzi, Fabien Heuze, Vân-Anh Huynh-Thu, Michel Journée, Thibaut Libert, Francis Maes, Adiamantios Marinakis, Alexandre Mauroy, Gilles Meyer, Frédéric Plumier, Laurent Poirrier, Pierre Sacré, Alain Sarlette, François Schnitzler, Olivier Stern, Laura Trotta and many other colleagues and friends from Montefiore that I forgot to mention here. I also would like to thank the administrative staff of the University of Liège, and, in particular, Marie-Berthe Lecomte, Charline De Baets and Diane Zander for their help.

I also would like to thank all the scientists, non-affiliated with the University of Liège, with whom I have had interesting scientific discussions, among others: Spyros Chatzivasileiadis, Jing Dai, Daniel Kirschen, Jean-Claude Maun and Patrick Panciatici.

Many thanks to the members of the jury for carefully reading this dissertation and for their advice to improve its quality.

I am very grateful to the FRIA (Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture) from the Belgium fund for scientific research FRS-FNRS and to the University of Liège for granting me my PhD scholarship. I also acknowledge the support of the Belgian interuniversity attraction pole DYSCO (Dynamical Systems, Control and Optimization).

I would like to express my deepest personal gratitude to my parents and family for teaching me the value of knowledge and work, and of course, for their love.

Finally, very warm thanks go to my nearest and dearest, Raphaël, and to my smiling little Gabrielle for their so precious support.

Florence Fonteneau-Belmudes

Liège, November 2011.

Abstract

This thesis presents an approach for identifying a maximal number of dangerous contingencies in large scale power system security assessment problems with bounded computational resources.

The method developed in this work relies on the definition of an objective function associating to each contingency a real value that quantifies its severity for the security of the system, this value being greater than or equal to a given threshold only for dangerous contingencies. The value of this function for a given contingency is computed from the result of a security analysis executed on the post-contingency configuration.

The framework we propose for identifying dangerous contingencies is derived from an algorithm from the optimization literature so as to find, with a given number of evaluations of the objective function, a maximal number of contingencies whose value of this function exceeds the adopted threshold. This approach performs successive samplings of the space gathering all the contingencies, and exploits the information contained in each of these samples in order to direct the subsequent sampling process towards contingencies with high values of the objective function. Our algorithm is first introduced in the case where the search space is a Euclidean space. Then we propose an extension of this approach to the more common case where the search space is discrete, thanks to a procedure allowing to embed a discrete contingency space in a Euclidean space, over which a metric is defined.

The efficiency of the developed method is evaluated on several case studies: an $N-3$ analysis of a benchmark test system, the IEEE 118 bus test system, and $N-1$ and $N-2$ studies of a real system, the Belgian transmission network.

Afterwards, we consider the case where several of these iterative sampling algorithms are available. Assuming that these algorithms are executed sequentially, we propose two different strategies for selecting on-line which of them to execute at the next step in order to identify as many dangerous contingencies as possible, while still respecting the given computational budget.

We finally provide an adapted version of the developed iterative sampling algorithm allowing to estimate the probability of occurrence of a dangerous contingency and the number of dangerous contingencies in a discrete search space.

Résumé

Cette thèse présente une approche permettant d'identifier un nombre maximal de contingences dangereuses dans des problèmes d'analyse de sécurité de réseaux électriques de grande taille lorsque les ressources informatiques disponibles sont bornées.

La méthode développée dans ce travail requiert la définition d'une fonction objectif associant à chaque contingence un nombre réel qui quantifie sa sévérité vis-à-vis de la sécurité du système, cette valeur étant supérieure ou égale à un seuil donné seulement pour les contingences dangereuses. La valeur prise par cette fonction pour une contingence donnée est calculée à partir du résultat d'une analyse de sécurité effectuée dans la configuration post-contingence.

L'approche proposée dans ce manuscrit pour identifier les contingences dangereuses est inspirée d'un algorithme d'optimisation et permet de trouver, en évaluant la fonction objectif un nombre limité de fois, un nombre maximal de contingences dont la valeur de cette fonction est supérieure ou égale au seuil fixé. Cette approche tire des échantillons successifs dans l'espace de contingences et exploite l'information qu'ils contiennent pour orienter le tirage suivant vers des contingences dont la valeur de la fonction objectif est élevée. La méthode développée est définie en premier lieu pour un espace de recherche continu. Elle est étendue dans un second temps au cas, plus courant, dans lequel cet espace est discret, grâce à une procédure permettant d'incorporer un espace de contingences discret dans un espace continu doté d'une métrique.

La méthode développée est ensuite mise en oeuvre dans plusieurs situations : une analyse $N-3$ d'un système de test de référence, le réseau à 118 noeuds de l'IEEE, et des analyses $N-1$ et $N-2$ d'un système réel, le réseau de transmission belge.

Ce manuscrit traite également le cas dans lequel plusieurs instances de cet algorithme d'échantillonnage itératif sont disponibles. En considérant que ces algorithmes sont appelés de manière séquentielle, deux stratégies sont proposées pour choisir au fur et à mesure lequel d'entre eux exécuter afin d'identifier autant de contingences dangereuses que possible, tout en respectant le budget de calcul fixé.

Pour finir, l'algorithme d'échantillonnage itératif mis au point dans cette thèse est adapté afin d'estimer la probabilité d'occurrence d'une contingence dangereuse ainsi que le nombre de contingences dangereuses dans un espace de contingences discret.

Contents

1

Introduction

This chapter introduces the field of power system security assessment. The problem addressed in this thesis is then motivated and precisely stated. Finally, a short summary of the different contributions exposed in the following chapters is provided.

1.1 Power system security assessment

1.1.1 Power systems

Electric power systems are one of the greatest human realizations. As our societies strongly depend on them, they have always benefited from advanced technological innovation. Even if their size and equipments may differ from one system to the other, they always have the same structure: they are made of generators of electricity, loads and transmission equipments carrying the power from the former to the latter.

The **generators** are mostly synchronous machines, which convert the mechanical energy supplied to a turbine's rotor into electrical energy supplied to the network. Different sources of energy (e.g., fossil, nuclear, hydraulic, wind-borne, ...) can be used to spin turbine's rotors. All the generating stations of a network work at the same nominal frequency (usually 50 or 60 Hz).

The **loads** range from industrial machinery to household appliance. They all need to be supplied with a frequency and voltage level standing in a tight range around their nominal values.

The **transmission equipments** are split between the transmission system, interconnecting the major generating stations and load centers at a high voltage level (typically, 69 kV and above), and the distribution system, which transfers power to the individual customers at a lower voltage level (usually up to 34.5 kV).

Generation and transmission facilities mostly use three-phase alternative current equipments, whereas the loads are either three-phased (usually the industrial ones) or single-phased (usually the commercial and residential ones), and are in this case roughly distributed among the phases in order to keep the imbalance between the three of them acceptable.

In a power system, the power supply is expected to meet the constantly changing demand while fulfilling several requirements such as the tight control of its frequency and voltage level. These requirements are achieved thanks to a wide variety of control actions, taken either locally on individual system elements or more globally by the Transmission System Operators (TSOs), who are responsible for the quality of the power supply.

1.1.2 Power system security

The operating conditions of a power system can be classified into different states, which reflect the level of security of the system and determine the appropriate control actions

to take. Dy Liacco defined in [1] three classes of power system states such that a power system is always operating in either one of them. His power system state diagram has been enriched in [2], where two new classes of states are introduced. The resulting diagram is represented on Figure 1.1. These power system "states" can be described as follows:

- in the so-called **normal** state, all the system variables are within the normal range and the system is secure with respect to the set of relevant contingencies likely to occur. The control objective in the normal state is to keep voltage levels and frequency close to their nominal values;
- in the **alert** state, all the system variables are still within the acceptability limits but close to them, and a contingency may lead the system to the emergency state or even to the *in extremis* state. Preventive control actions should be taken to bring the current operating conditions to the normal state;
- in the **emergency** state, some components' operating limits are violated but the system is still intact. The corrective control objective is to relieve system stress and go back to alert or normal conditions;
- the ***in extremis*** state is characterized by the disintegration of the entire system into smaller islands, or by a complete blackout. Emergency control actions aim at saving as much of the system as possible from a widespread blackout if it is still feasible;
- in the **restorative state**, restorative control actions are being taken to reconnect lost generators and restore load, so as to bring the system back to either alert or normal operating conditions.

Nowadays, the trend is that power systems operate more and more in the alert state.

1.1.3 Security assessment

Power system security assessment consists in analyzing the ability of the system to withstand any likely changes in its current operating conditions, i.e. to remain in the normal or alert operating state.

The events triggering changes in the operating conditions of a system are named contingencies. Among the very vast range of such events are for instance transient faults, equipment outages but also changes in the load and generation patterns, as well as human errors of the operators. The notion of contingency can also be used to model the uncertainties on the future generation and load patterns.

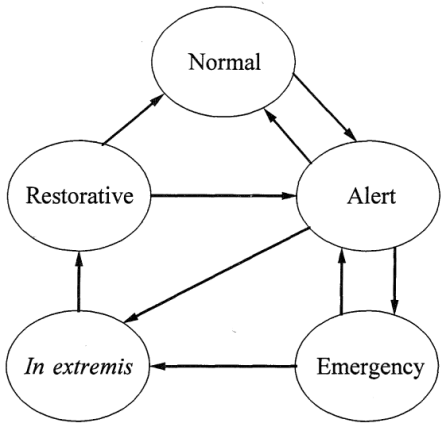


Figure 1.1: Power system state diagram defined in [2].

In practical terms, a security assessment procedure considers a given set of potential contingencies and evaluates, thanks to numerical simulations, to which operating state the post-contingency steady-state operating conditions correspond to, if a post-contingency steady-state exists. If a contingency is found out to lead the system towards the emergency or *in extremis* state, preventive control actions should be taken so that the system goes back to the normal state. The different numerical tools that can be used for simulating the effect of a contingency on the studied power system are presented hereafter.

- **Transient angle stability analyses.** Transient stability is the ability of a power system to maintain synchronism after the occurrence of a disturbance like a fault on transmission equipments, loss of generation or loss of load (see [3]). A transient stability analysis requires to perform time-domain simulations of the system's dynamic response after the occurrence of such a disturbance. This procedure implies to solve a set of mixed algebraic and ordinary differential equations, which are strongly nonlinear.

- **Voltage stability analyses.** Voltage stability is the property of a power system which enables it to remain in a state of equilibrium (i.e., with acceptable voltages at all buses) under normal operating conditions and to regain an acceptable state of equilibrium after a disturbance (see [4]). A voltage stability analysis can be

carried out either using static methods, allowing in particular to evaluate how close the system is to voltage instability, or time-domain simulations in case when it is necessary to study voltage collapse dynamics (see [5]).

- **Load-flow calculations.** When focusing only on the post-contingency steady-state, the voltage magnitudes and angles for all buses of the system as well as the real and reactive power flows in the transmission lines can be computed thanks to an AC power flow algorithm (see [6]). This algorithm iteratively solves a set of nonlinear equations including all the nodal current balance equations of the system. In some cases (e.g., when the user only needs to compute the real power flows in the transmission lines, or when the voltage variations are not too important), a DC power flow algorithm can be used (see [7]). It solves in a non-iterative way a simplified and linear model of the AC system.

Security assessment is performed at different time scales, both ahead of time for operation planning and in real-time operation. It mainly allows to identify the set of contingencies which would bring the security level of the system below an acceptable threshold. This set is afterwards used by the operator to design adapted preventive and corrective control actions.

The set of potential contingencies considered in power system operation is typically the set of what TSOs call $N-k$ contingencies, i.e. all the events consisting in the sudden loss of k transmission equipments among the N available ones. When the value of k is rather small (in practice, it is commonly set equal to 1 or 2), the potential contingencies are analyzed individually: each of them is simulated in order to assess its effect on the security of the system. Even in large scale systems, this problem remains perfectly tractable. When k takes higher values, the size of the set of potential $N-k$ contingencies can be really huge, especially in wide interconnected systems which comprise a very high number of equipments, and considerable computational resources would be necessary to simulate them one by one.

As mentioned previously, the notion of contingency can also refer to changes in the generation profile, which can happen at every moment and with various amplitudes due to the increasing penetration of renewable energies among generation patterns (in particular, wind power and solar energy). The number of potential contingencies grows exponentially with the length of the considered time horizon and their exhaustive screening would therefore require computational resources that also grow exponentially.

In situations where the available amount of computational resources does not allow an exhaustive screening of the contingency space within a reasonable amount of time, even with parallel computation, the system operators usually choose to focus on the $N-$

k contingencies that seem a priori more likely to occur (named credible contingencies in [8]). They also generally include in their studies the contingencies, whatever their probability, that have already affected the system in the past.

This thesis proposes an alternative to this latter procedure of selection of the contingencies to be analyzed which is not based on the probability of occurrence of the contingencies, but rather on information on the similarities between contingency severities directly extracted from the search space.

1.2 Problem addressed in this thesis

1.2.1 Studied setting

We focus in this thesis on large scale power system security assessment problems, like $N-k$ analyses when k takes a large value, or security assessment problems in which the contingencies model the wide variety of the potential generation patterns that it is possible to observe. In such problems, the set of potential contingencies (named *contingency space* in the following and denoted by $\mathscr{X}$) is often too large to be screened exhaustively in a reasonable amount of time with the available computational resources.

We specifically address large scale security assessment problems submitted to a constraint on the available computational resources, and propose an alternative to the exhaustive screening of the contingency space for identifying a maximal number of dangerous ones, i.e. those driving the system to unacceptable operating conditions, while using a given amount of computational resources. In a few words, the developed algorithm iteratively draws samples of contingencies from the contingency space and analyses at each step the properties of the current sample to update the parameters of the sampling mechanism and orient it towards the most dangerous contingencies.

1.2.2 Assumptions

The research presented in this manuscript relies on the definitions of the notions of contingency severity, dangerous contingencies and computational resources detailed hereafter.

1.2.2.1 Contingency severity: objective function

First, we assume that there exists a real-valued function defined on the contingency space that quantifies the effect of each contingency on the operating conditions of the

system. Such a "severity" function is named *objective function* in the following and denoted by $O : \mathscr{X} \to \mathbb{R}$. It is built from the result of the numerical simulation of each post-contingency steady-state, and more precisely based on the system state variables. The evaluation of the objective function for a given contingency thus requires to run a security analysis in order to calculate the post-contingency operating conditions of the system.

As it will be explained later, the choice of the involved state variables depends on the performed study: for instance, the value of the objective function can reflect the loading rate of one or several transmission lines, or the amount of available reactive power reserve in the post-contingency steady-state.

1.2.2.2 Dangerous contingencies

A contingency is classified as dangerous or non-dangerous according to its value of the objective function: the *dangerous contingencies* are the contingencies for which this value exceeds a threshold defined by the user. This threshold, denoted by $\gamma \in \mathbb{R}$ in the following, is defined according to the acceptability limits of the operating conditions of the system and, naturally, is closely linked with the adopted objective function. For instance, if the objective function focuses on one single transmission line and is set equal to the value of the current flowing in this line, a logical choice for the value of γ is the maximal current the line is admitted to carry. Note that, in large scale power system security assessment problems, the dangerous contingencies are usually rare with respect to the non-dangerous ones.

1.2.2.3 Computational resources

What we refer to as computational resources in this thesis is a fixed budget in terms of CPU time.

We will assume in the following that the CPU time required by all the operations performed by the approach proposed in this work (e.g., drawing points from the contingency space or computing the parameters of a sampling distribution) is negligible with respect to the amount of CPU time required to evaluate the objective function. Indeed, this evaluation requires running a security analysis for a contingency, even with a simple analysis tool such as a DC load-flow, turns out to be computationally much more demanding than the operations we make for drawing points and fitting distributions. We will also assume that the security analyses have more or less the same running time, whatever the contingency. This assumption allows us to translate the computational resources into a number of contingencies analyses that can be carried

out (either parallelly or sequentially), which also corresponds to the maximal number of evaluations of the objective function that can be performed.

1.2.3 Problem statement

The objective of the research work presented in this thesis is to identify, with bounded computational resources, as many dangerous contingencies as possible in a large scale power system security assessment problem.

As explained in the previous subsection, a contingency is defined as dangerous if its value of the objective function exceeds a user-defined threshold γ. According to this practical definition of dangerous contingencies, the problem statement can be rephrased as:

Identify, with bounded computational resources (expressed here as a given number of evaluations of the function O), a maximal number of contingencies x such that $O(x) \geq \gamma$.

Note that our approach does not pretend to identify all the dangerous contingencies in the considered problem, which could be done with an exhaustive screening of all potential contingencies. Rather it aims at taking better advantage of the given number of contingency analyses that can be carried out and identifying more dangerous contingencies than if the contingencies to be analyzed were drawn from the contingency space using a classical Monte Carlo sampling process.

1.2.4 Proposed procedure

As formulated in the previous subsection, the dangerous contingency identification problem addressed in this thesis can be seen in some ways as an optimization problem. In particular, if the parameter γ is set equal to the maximal value of the objective function, our problem can be parented to a classical optimization problem – apart from the fact that we do not only want to identify one maximum of the objective function but as many as possible, ideally all of them.

In practice, it is more likely that γ takes a lower value than the maximum of the objective function, which makes our problem even more different from an optimization problem. We however propose to address it using a method derived from the field of optimization (more specifically, from Derivative-Free Optimization), by exploiting the fact that these algorithms come across a significant number of contingencies such that $O(x) \geq \gamma$ while searching for a maximum of the objective function.

We propose in this work an iterative sampling framework inspired from these algorithms to efficiently identify contingencies with a value of the objective function greater than or equal to γ. This approach performs successive samplings of the contingency space with probability distributions that evolve along the iterations. Each time a new sample is drawn, it is analyzed by evaluating the objective function for each of its elements so as to extract the information it contains. Depending on the properties of the elements of this sample, the parameters of the sampling distribution according to which the following sample will be drawn are adjusted in order to give strong preference to the elements with high values of the objective function.

1.2.5 Related approaches

Several approaches to large scale power system security assessment when the available computational resources are bounded (in other words, approaches proposing alternatives to an exhaustive screening of wide contingency spaces) have already been published in the literature.

These methods usually encompass a prior filtering process based on some light computations or some prior knowledge of the network, followed by a detailed analysis of the selected contingencies in order to evaluate the consequences they would have on the security of the system.

It is for instance what is done by the unified approach to transient stability contingency filtering, ranking and assessment introduced in [9]. Using the SIngle Machine Equivalent (SIME) method (described in [10] and [11]), this procedure first screens the contingencies and assesses them approximately to discard the most stable ones, and then assesses the potentially interesting ones in a detailed way. Thanks to this filtering process, only little CPU time is spent to analyze stable contingencies and the computational resources are essentially used for accurately assessing the less stable ones.

A different pre-processing of the contingency space is used in [12] and [13], which propose a framework based on event trees to identify contingencies that would lead to large system disturbances due to voltage collapse. The considered contingency space comprises in particular all the potential system failures modes based on incorrect protection operation, and is therefore very wide. The sequences of events that may lead to large system disturbances are studied by building event trees, which represent the possible disturbance developments in a given base case. In order to limit the size of such trees, a vulnerability region is associated to each possible initial fault location and only sequences including events happening in the vulnerability region of the equipment affected by the initial fault are developed. All the other potential sequences of events are excluded from the analysis, based on the assumption that it is a priori less likely that they would lead to large system disturbances.

The approach for identifying dangerous contingencies as rare events in large contingency spaces presented in [14] and [15] also makes some assumptions to restrict the subset of contingencies covered by the study: based on the rare event approximation (see [16]) and on existing statistics, contingencies with a priori too low probability of occurrence (with respect to a user-specified threshold) are cut off. The total number of considered contingencies, from which the proposed algorithm extracts a list of high risk contingencies, is thus limited to a number that is linearly proportional to the scale of the system.

In the field of static security assessment, [17] has proposed a two-stage procedure for identifying dangerous $N-k$ line tripping contingencies. In this approach, a prior selection of a subset of candidate lines to which the security of the system is sensitive allows to limit the number of $N-k$ contingencies to be analyzed. The screening and selection process performed in the first stage of this approach relies on both graph partitioning and optimization methods, and use simplified models so that the analysis remains tractable in spite of the size of the contingency space. The second stage consists in a deeper analysis (with detailed models) of the $N-k$ contingencies involving the selected candidate lines, which are much less numerous than the potential $N-k$ contingencies.

All these approaches overcome the combinatorial aspect of large scale power system security assessment problems by bringing initial restrictions to the set of considered potential contingencies, according to some simplified simulations, or to their a priori probability of occurrence or their a priori consequences. To the contrary, our framework draws samples from the whole contingency space, and uses the observed characteristics of the contingencies contained in these samples (i.e., the value of the objective function for each contingency they contain) to determine on which areas to focus in the following iterations. As we will see, this procedure is efficient enough to come across dangerous contingencies while using bounded computational resources. Moreover, it can allow to identify contingencies that would be excluded from the study by the adopted filtering criteria, but that are however dangerous for the system.

1.3 Main contributions of this work

The contributions exposed in this dissertation are the following:

- The main contribution of this thesis is the development of a framework using iterative sampling to search in large contingency spaces for identifying a maximal number of dangerous contingencies with bounded computational resources.

This contribution is briefly detailed hereafter in Section 1.4 and fully reported in Chapter 2.

- The second contribution of this work introduces several ways for embedding a discrete contingency space in a Euclidean space over which a metric is defined. This contribution is summarized in Section 1.5 and detailed in Chapter 3.
- As third contribution of this thesis, we propose strategies to select on-line which iterative sampling algorithm to execute in the case where several of them are available, so as to identify as many dangerous contingencies as possible while respecting a given computational budget. This contribution is briefly described in Section 1.7 and fully developed in Chapter 5.
- The fourth contribution of this thesis is an algorithm inspired from the cross-entropy method for rare-event simulation for estimating the probability of occurrence of the set of dangerous contingencies and estimating its size if the contingency space is discrete. This contribution is summarized in Section 1.8 and presented in Chapter 6.
- As fifth contribution, we provide extensive simulation results throughout the whole manuscript and especially in Chapter 4, where different case studies are presented. A detailed overview of these case studies is given in Section 1.6.

Short technical summaries of the different chapters of the present dissertation are provided in the following sections of this introduction.

1.4 Chapter 2: an iterative sampling approach based on derivative-free optimization methods

This chapter begins with an analysis of the similarities of the problem addressed in this thesis with an optimization problem. Derivative-free optimization methods are introduced and a particular instance of an iterative sampling method for derivative-free optimization, the cross-entropy method, is presented. The framework we propose in this thesis for efficiently identifying dangerous contingencies in a large scale security assessment problem is an iterative sampling approach derived from this latter method. A tabular version of this algorithm as well as a detailed explanation are provided, followed by a further version adapted to the case where the available computational resources are bounded. These two algorithms are written in the case where the contingency space is a Euclidean space.

Before providing a preliminary and simple illustration of this approach, we discuss the role and form of the objective function and present different ways to define it.

1.5 Chapter 3: embedding the contingency space in a Euclidean space

While our iterative sampling approach is first introduced in Chapter 2 for investigating Euclidean contingency spaces, the contingency spaces considered in most power system security assessment problems are discrete. In order to apply the developed iterative sampling process, it is hence necessary to embed the contingency space in a Euclidean space, over which a metric is defined. We propose in this chapter several ways to do so, depending on the nature of the considered contingencies and on the properties of the considered electricity transmission networks.

1.6 Chapter 4: case studies

This chapter report results obtained when applying this approach to different security analysis problems and different networks. Section 4.1 presents results obtained on the IEEE 118 bus test system for an $N-3$ analysis. Section 4.2 collects results of simulations performed on the Belgian transmission system, for an $N-1$ security analysis when the objective function is based on a local criterion (the loading rate induced by each contingency on a specific targeted transmission line). Section 4.3 finally reports the results of an $N-2$ analysis, also carried out on the Belgian transmission system but using here a global objective function (the maximal loading rate induced by a contingency on any transmission line).

1.7 Chapter 5: on-line selection of iterative sampling algorithms

We consider in this chapter that several iterative sampling algorithms have been built on the same dangerous contingency identification problem. We propose two strategies for combining them in order to identify as many dangerous contingencies as possible while respecting a given computational budget.

1.8 Chapter 6: estimating the probability and cardinality of the set of dangerous contingencies

The first chapters of this thesis focus on the identification of a maximum number of dangerous contingencies under computational constraints. We may also imagine that, in some settings, the system needs to compute the probability of occurrence of a dangerous contingency and, only if this probability is above a certain threshold, to decide to take preventive control actions. This could be done in principle by identifying all the dangerous contingencies and summing their probabilities. However, this problem can also be seen as a rare-event problem and specific algorithms for tackling this problem could be used. This chapter describes one of these algorithms and its application to the problem of estimating the probability of occurrence of a dangerous contingency under computational constraints. Its also explains how the obtained estimate can be used to compute the number of dangerous contingencies in the case where the contingency space is discrete.

1.9 Chapter 7: conclusion

Chapter 7 finally discusses the methodology and the obtained results, proposes future work and presents concluding remarks.

1.10 Appendix A: pseudo-geographical representations of power system buses by multidimensional scaling

This appendix proposes new ways to visualize power systems, based not only on the geographical coordinates of the equipments but also on some physical information related to them. The pseudo-geographical representations thus created, that were derived from our research about the definition of a metric on the contingency space (presented in Chapter 2), can help to gain insights into the physical properties of the network.

1.11 List of publications

The work presented in this thesis has already been published in several articles:

- "Consequence driven decomposition of large scale power system security analysis." F. Fonteneau-Belmudes, D. Ernst, C. Druet, P. Panciatici and L. Wehenkel. In *Proceedings of the 2010 IREP Symposium - Bulk Power Systems Dynamics and Control - VIII*, Buzios, Rio de Janeiro, Brazil, August 1-6, 2010. (8 pages).

- "Pseudo-geographical representations of power system buses by multidimensional scaling." F. Fonteneau-Belmudes, D. Ernst and L. Wehenkel. In *Proceedings of the 15th International Conference on Intelligent System Applications to Power Systems (ISAP 2009)*, Curitiba, Brazil, November 8-12, 2009. (6 pages).

- "A rare event approach to build security analysis tools when $N-k$ ($k>1$) analyses are needed (as they are in large scale power systems)." F. Fonteneau-Belmudes, D. Ernst and L. Wehenkel. In *Proceedings of the 2009 IEEE Bucharest PowerTech Conference*, Bucharest, Romania, June 28 - July 2, 2009. (8 pages).

- "Cross-entropy based rare event simulation for the identification of dangerous events in power systems." F. Belmudes, D. Ernst and L. Wehenkel. In *Proceedings of the 10th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS-08)*, Rincon, Puerto Rico, May 25-29, 2008. (7 pages).

2

An iterative sampling approach based on derivative-free optimization methods

We first analyze in this chapter the similarities of the problem addressed in this thesis (under the formulation proposed in the introduction) with an optimization problem. This analysis is followed by an introduction to the general concepts behind derivative-free optimization methods, which would be used to solve the optimization problem our problem stands the closest to. A detailed description of one such algorithm, the cross-entropy method, follows. Afterwards, we present an iterative sampling algorithm derived from this method for solving the problem addressed in this thesis and propose a way to take into account the fact that the available computational resources are bounded. The properties of the objective function are then discussed and several examples are provided for defining it so as to adapt our approach to the power system security assessment problem at hand. To close this chapter, the proposed approach is illustrated on a very simple security assessment problem.

For clarity reasons, the approach introduced in this chapter is based on the assumption that the contingency space is Euclidean. The – more common – case where it is discrete will be addressed in Chapter 3.

2.1 Comparison with an optimization problem

An optimization problem is usually stated as follows:
given a search space $\mathscr{X}$ and a real-valued function $f : \mathscr{X} \to \mathbb{R}$, identify an element $x_0 \in \mathscr{X}$ such that $f(x_0) \geq f(x)\ \forall x \in \mathscr{X}$.

As a reminder, the problem addressed in this thesis is the following:
given a search space $\mathscr{X}$, an objective function $O : \mathscr{X} \to \mathbb{R}$ and a real number γ, identify a maximal number of points $x \in \mathscr{X}$ such that $O(x) \geq \gamma$ with bounded computational resources (expressed here as a given number of evaluations of the function O).

This latter problem shares many similarities with the typical formulation of an optimization problem. The contingency space $\mathscr{X}$ can be seen as the search space, and the objective function O plays the exact same role as the function f (which is also named "objective function" in the field of optimization). However, as highlighted in the introduction, we do not want in our problem to identify only one point of the search space maximizing the objective function, but all the points x such that $O(x) \geq \gamma$. Note that setting γ equal to the maximum of the function O would make the problem equivalent to an optimization problem whose goal would be to identify as many maxima of the objective function as possible and not only a single one.

The specificities of the "optimization-like" problem this work aims at solving are the following. First, the size of the search space is combinatorial in the large scale security assessment problem we address.

Second, while the objective function can be defined in many different ways according to the performed study, we want to propose a generic framework for solving the problem stated in Chapter 1.2.3 whatever the contingency space and objective function at hand. The objective function can therefore not be expected to have a particular structure, and especially to be linear or convex, so that the developed framework can be applied in the very frequent (but not systematic) case where this function is nonlinear and nonconvex.

Moreover, no derivative of this function is available and the problem can only be solved using the pairs $(x, O(x))$ composed of a contingency and its value of the objective function.

Finally, the constraint our problem is subjected to is the given bound on the available computational resources, which is expressed as a maximal number of evaluations of the objective function that can be performed. This implies that the number of different pairs $(x, O(x))$ that can be formed is restricted.

2.2 Derivative-free optimization methods

Optimization problems can either be solved by evaluating Hessians (like in Newton's method), or by computing gradients (as done, among others, by Quasi-Newton's method or gradient descent), or based only on the values taken by the objective function when no derivative of this function is available (using for instance pattern search methods). We will focus in this chapter on this latter type of optimization methods (named derivative-free optimization methods), which are adapted to solve the optimization problem our dangerous contingency identification tends to be similar to, i.e. an optimization problem where only values taken by the objective function for different points of the search space can be used to search for a maximum of this function.

Derivative-Free Optimization (DFO) methods either build a model of the objective function based on samples of its values, as the interpolation methods presented in [18], or they directly exploit sets of values of the objective function without building an explicit model and iteratively try to improve a candidate solution of the considered optimization problem (see [19]). Such "direct" methods are usually called metaheuristics, or simply iterative sampling methods (which will be the denomination adopted in this thesis). They are often considered as methods of last resort, that is, applicable to problems where the search space is large, complex and poorly understood. Note that these are precisely the properties of the problem addressed in this thesis, in which the search space is large by definition and can be considered as poorly understood in the sense that the profile of the objective function over this space is not known a priori.

Many iterative sampling methods have been proposed in the literature, such as genetic algorithms ([20]), evolution strategies ([21]), distribution estimation methods ([22]) and the cross-entropy method ([23]).

In the following subsections, we first present the generic algorithmic behavior of iterative sampling methods and then focus on a particular DFO algorithm, from which we will derive a framework addressing the dangerous contingency identification problem considered in this thesis.

2.2.1 General principle of iterative sampling methods for derivative-free optimization

Iterative sampling algorithms navigate in the search space towards points with the highest values of the objective function. From an algorithmic point of view, these methods combine random sampling with an iterative process allowing to "learn" the best sampling scheme for the problem at hand. When applied to a classical optimization problem with search space $\mathscr{X}$ and objective function f, the different steps of such algorithms would be the following:

- define an initial sampling distribution over the search space;
- at each iteration:
  - generate a subset of potential solutions over the search space by using the current sampling distribution;
  - evaluate the objective function for each point in the current sample;
  - use the pairs $\{$point, value of the objective function$\}$ in the current sample so as to determine a new sampling distribution better targeting points with high values of the objective function;
- halt the iterative process when the computational resources have been exhausted, or when the current sample is sufficiently pure in terms of objective function distribution, or when the variations of some sample statistics have not changed significantly over a certain number of iterations;
- return the point with the highest value of the objective function encountered during the whole execution of the algorithm.

This algorithm generates over the iterations a sequence of sampling distributions defined on the search space which progressively target subsets of points with growing values of the objective function.

2.2.2 A particular instance of iterative sampling methods: the cross-entropy method

The cross-entropy method for optimization (see [23]) applied to a Euclidean search space proceeds as follows:

- define a hypothesis space of candidate sampling densities p_λ defined over $\mathscr{X}$ and indexed by a parameter vector λ. This space of distributions may be chosen in a problem specific way, for example by taking into account properties such as linearity, gaussianity or the possibility of multiple modes;
- set λ to its initial value λ_0 (λ_0 will typically be chosen so as to let the distribution p_{λ_0} cover the complete space $\mathscr{X}$);
- at each iteration i, draw a sample S_i of size s of points in $\mathscr{X}$ according to the current distribution defined by the current value λ_i (s is a parameter of the algorithm) and evaluate the value of the objective function f for each of these points;

- keep the subset S'_i of S_i corresponding to the $m < s$ best solutions (m is another parameter of the algorithm), i.e. to the highest values of the objective function;
- use the sample S'_i to determine a new value λ_{i+1}. λ_{i+1} is typically chosen such that the likelihood of the sample S'_i is maximal with respect to the selected space of distributions;
- stop when the chosen stopping criterion (e.g., no more computational resources available or no significant change in the variations of some sample statistics over the last iterations) is met;
- return the point of $\mathscr{X}$ with the highest value of the objective function among all samples S_i drawn over all the iterations that have been performed.

2.2.3 An interesting property of derivative-free optimization methods

While navigating through the search space towards a maximum of the objective function, iterative sampling methods as the cross-entropy method presented in the previous subsection come across points with increasing values of the objective function over the iterations.

To illustrate this property, we have implemented the cross-entropy algorithm on a simple optimization problem with $\mathbb{R}$ as search space, the function $f(x) = -0.1x^2 + 10$ as objective function, and parameters $s = 30$ and $m = 5$. The average values taken by the objective function in the successive samples drawn during a typical run of this algorithm are represented on Figure 2.1. The horizontal axis of this figure represents the number of the current iteration of the algorithm while each vertical bar represents the average plus and minus the standard deviation of the objective function in the sample of 30 points generated during a given iteration.

We observe that the average value of the objective function in each of these samples rapidly grows over the iterations to its maximal value (10) while its standard deviation decreases as rapidly, which shows that the samples of points drawn from $\mathbb{R}$ contain more and more points with high values of the objective function over the iterations.

We propose in this work to exploit this property so as to identify the contingencies with a value of the objective function greater than the threshold γ. To do so, we use exactly the same algorithmic structure as the cross-entropy method, and, after each evaluation of the objective function for a point drawn from the contingency space, we check if the obtained value is greater than or equal to γ. If it is the case, the corresponding contingency is stored in a set $\mathscr{X}_{dang}$ gathering the dangerous contingencies identified throughout the execution of the algorithm.

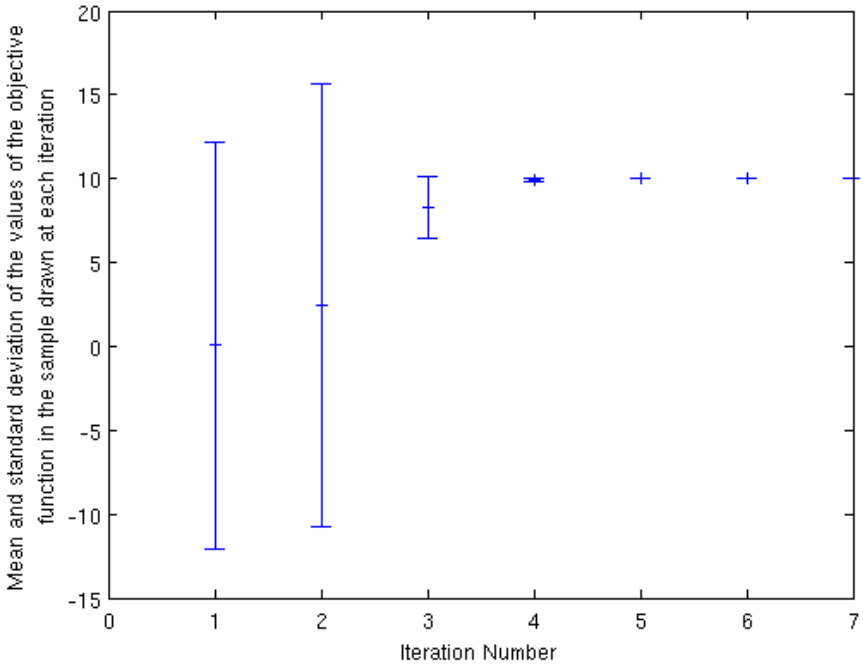


Figure 2.1: Evolution of the mean and standard deviation of the values taken by the objective function in the samples drawn over the iterations of a typical run of the cross-entropy algorithm. The problem addressed here is a simple optimization problem.

2.3 A basic iterative sampling algorithm for dangerous contingency identification

We propose in this section an iterative sampling algorithm, inspired from the cross-entropy method for identifying dangerous contingencies in a Euclidean space of dimension n. For the sake of clarity, the constraint on the available computational resources is not taken into account in this first algorithm – which is hence characterized as basic – and will be implemented in the comprehensive iterative sampling approach to be presented in the following section.

The inputs of this basic iterative sampling algorithm are a Euclidean contingency

space $\mathscr{X}$, an objective function O and a threshold $\gamma \in \mathbb{R}$ defining the dangerous contingencies, a contingency x being dangerous if $O(x) \geq \gamma$. It outputs a set of contingencies whose value of the objective function is greater than γ.

The algorithm uses as sampling distributions Gaussian laws defined on the search space $\mathscr{X}$ (denoted in the algorithm by $Gauss_{\mathscr{X}}(\cdot, \lambda_i)$). The parameters $\lambda_0 = [\mu_0, \Sigma_0]$ of the initial sampling distribution (μ and Σ refer to the mean and the covariance matrix of the distribution, respectively) are chosen such that the initial sampling distribution covers well the entire contingency space.

At each iteration i, a sample of s elements is drawn according to $Gauss_{\mathscr{X}}(\cdot, \lambda_i)$. Afterwards, the different values that the objective function takes over these contingencies are computed. The contingencies which lead to the m highest values of the objective function are then used to compute the mean and covariance matrix μ_{i+1} and Σ_{i+1} of the next sampling distribution. μ_{i+1} is thus set equal to the mean of the coordinates of the m best scoring contingencies, and Σ_{i+1} to their covariance matrix. Usually, in the cross-entropy framework (see [23]), the value of s is chosen one order of magnitude larger than the number of elements parameterizing the sampling distributions and the parameter m is chosen 10 to 20 times smaller than s.

The information extracted from the data sampled at each iteration i is stored as a set P_i of pairs (x, o) gathering a contingency x and the corresponding value $o = O(x)$ of the objective function.

Different stopping conditions can be chosen, as for instance checking if some statistics of the current sample are below a certain threshold or if the decreasing of these statistics is too slow. In this detailed algorithm, we will for illustrative purposes stop the algorithm when a specific number of iterations i_{max} has been reached.

The fully specified version of this basic iterative sampling algorithm is provided in a tabular form in Figure 2.2.

To illustrate this description of the proposed algorithm, the series of figures (Figures 2.3 to 2.9) inserted after the fully specified version of the algorithm presents the different steps of an execution of the iterative sampling algorithm. In the toy problem treated here, the contingency space is a rectangular subpart of the plane, the objective function is a two-dimensional Gaussian defined on this space, and there is only one dangerous contingency (the one corresponding to the maximum of the objective function on the contingency space).

Problem definition: a Euclidean contingency space $\mathscr{X}$, an objective function $O : \mathscr{X} \to \mathbb{R}$ and a threshold $\gamma \in \mathbb{R}$.

Algorithm parameters: the parameters $\lambda_0 = [\mu_0, \Sigma_0]$ of the initial Gaussian sampling distribution, the size s of the sample drawn at each iteration, the number m of best solutions chosen at each iteration and the maximal number i_{max} of iterations to be done.

Output: a set $\mathscr{X}_{dang}$ of elements of $\mathscr{X}$ such that $O(x) \geq \gamma$.

Algorithm:

Step 1. Set $i = 0$, set $\mathscr{X}_{dang}$ to the empty set.

Step 2. Set S_i, P_i and S'_i to empty sets.

Step 3. Draw independently s elements of $\mathscr{X}$ according to the probability distribution $Gauss_{\mathscr{X}}(\cdot, \lambda_i)$ and store them in S_i.

Step 4. For every element $x \in S_i$, compute $o = O(x)$ and add the pair (x, o) to P_i.

Step 5. Identify in P_i the pairs for which $o \geq \gamma$ and set their x values in $\mathscr{X}_{dang}$ if they are not already in it.

Step 6. Identify in P_i the m pairs with the highest values of o and set their x values in S'_i.

Step 7. If $i < i_{max} - 1$, set $\mu_{i+1}[j] = \frac{1}{m} \sum_{x \in S'_i} x[j]$ for $j = 1, \ldots, n$,

$\Sigma_{i+1} = \frac{1}{m} \sum_{x \in S'_i} (x - \mu_{i+1})(x - \mu_{i+1})^T$ and $\lambda_{i+1} = [\mu_{i+1}, \Sigma_{i+1}]$. Set $i \leftarrow i + 1$ and go to **Step 2.**

Else, go to **Step 8.**

Step 8. Output $\mathscr{X}_{dang}$ and stop.

Figure 2.2: A basic iterative sampling algorithm for identifying the elements such that a function $O : \mathscr{X} \to \mathbb{R}$ exceeds a threshold γ when the space $\mathscr{X}$ is Euclidean.

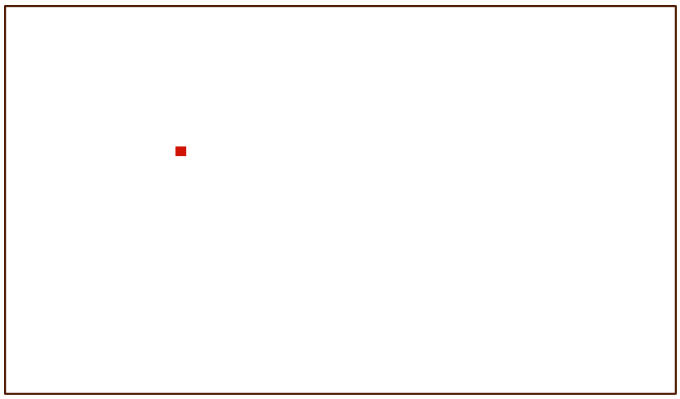


Figure 2.3: Contingency space (every little square represents a different contingency). The targeted dangerous contingency is represented by the red square.

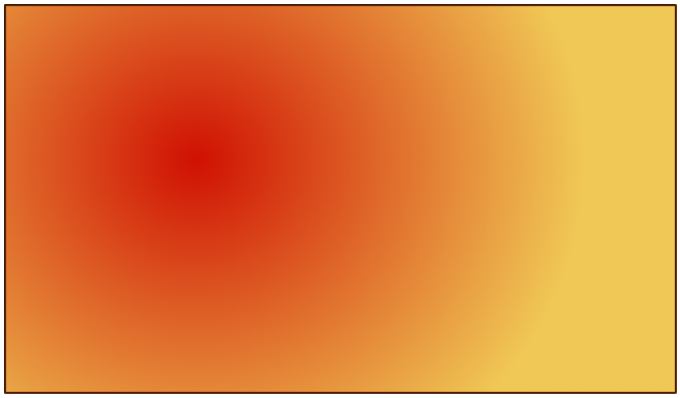


Figure 2.4: Profile of the values of the objective function. Color scale: from yellow (lowest values) to red (highest values).

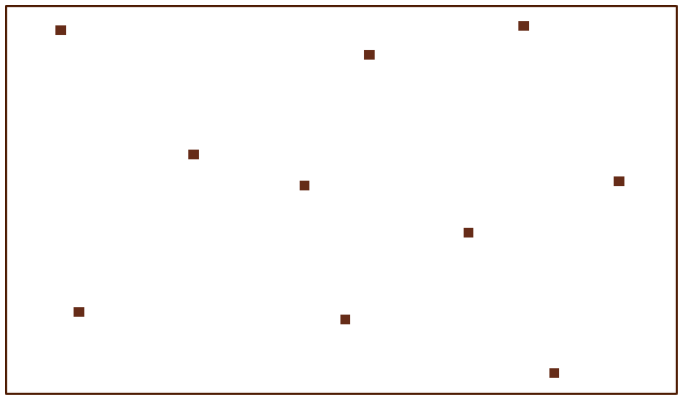


Figure 2.5: First iteration, sample drawn from the contingency space.

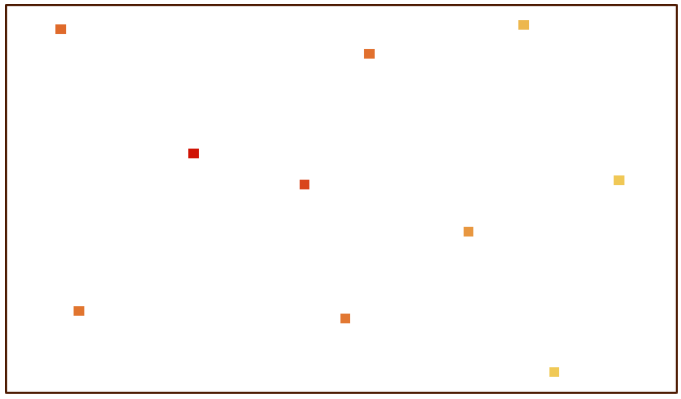


Figure 2.6: First iteration, evaluation of the value of the objective function for the points contained in the sample.

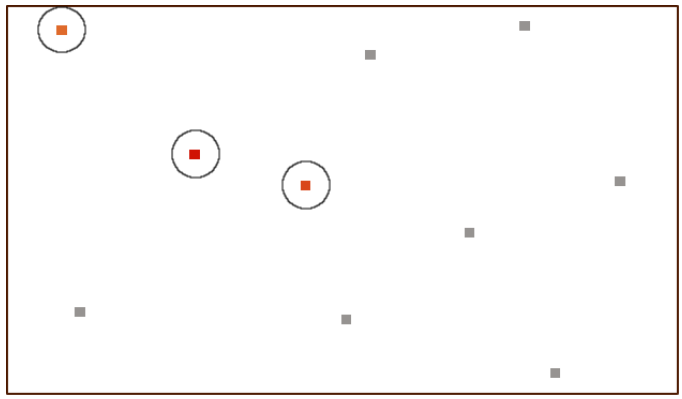


Figure 2.7: First iteration, selection of the points with the highest values of the objective function. Their coordinates are then used to update the current sampling distribution to better target the area they are located in.

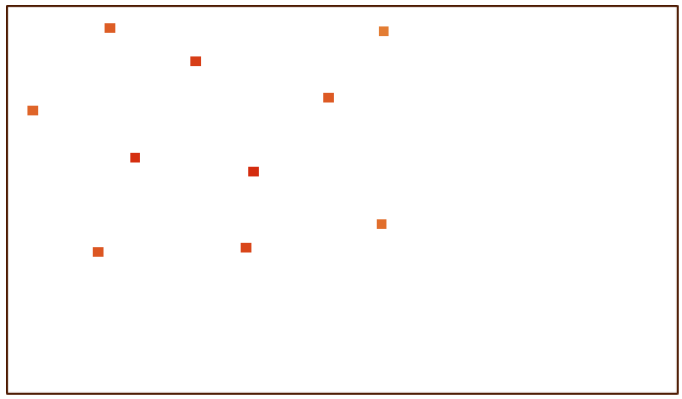


Figure 2.8: Second iteration, new sample drawn from the contingency space. We oberve that, after one iteration, the sampling process has already been oriented towards the dangerous contingency.

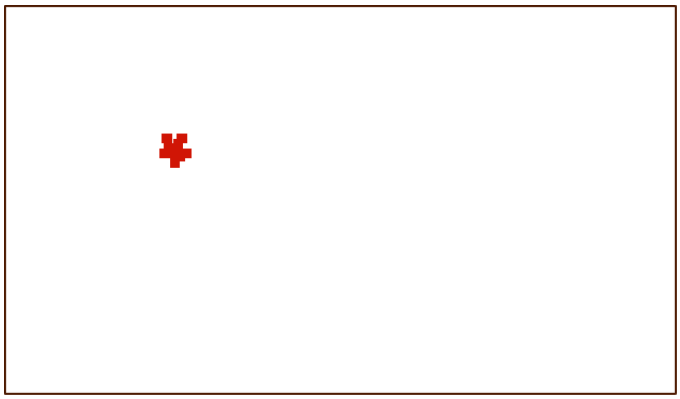


Figure 2.9: Last sample drawn from the contingency space before stopping the algorithm. The current sampling distribution is now centered around the sought contingency.

2.4 A comprehensive iterative sampling approach for dangerous contingency identification with bounded computational resources

2.4.1 A fully specified algorithm

We provide in the following a comprehensive iterative sampling algorithm to identify dangerous contingencies while respecting a given computational budget. As already explained in the introduction, the amount of available computational resources is expressed as a maximal number of times it is possible to evaluate the objective function, or, in other words, as a maximal number of contingencies that can be analyzed. We propose to implement this constraint by bringing the following changes to the basic iterative sampling algorithm that was presented in the previous section:

- first, an additional stopping condition is introduced so as to end the execution of the algorithm when the available computational resources are exhausted. More precisely, the algorithm stops analyzing the contingencies contained in the current sample when the amount of remaining computational resources is equal to zero. Then it checks if there are some dangerous contingencies among the contingencies in the current sample that have been analyzed before the exhaustion

of the computational resources. These dangerous contingencies are sent to the set $\mathscr{X}_{dang}$ and the algorithm finally stops.

- second, if the computational budget has not been exhausted when the basic iterative sampling algorithm stops by itself, this algorithm is re-run (with a different "random number" seed) until this budget is reached so as to make the best use of the available resources. In the case when a new run (named "sub-run" in the following) of the algorithm is performed, the set $\mathscr{X}_{dang}$ of dangerous contingencies identified is of course not re-initialized.

- finally, the amount of available computational resources is decremented by 1 each time the objective function is evaluated for a contingency and the algorithm checks before any new evaluation of the objective function if there are still some computational resources available.

A tabular fully specified version of the comprehensive iterative sampling approach to dangerous contingency identification resulting from these changes is provided hereafter in Figure 2.10.

2.4.2 Use of parallel computing

In order to improve the speed of execution of the algorithm, the values of the objective function for all the contingencies contained in the sample drawn from the contingency space at each iteration can be computed parallelly, instead of sequentially as proposed in Figure 2.10. In this case, it is necessary to declare the amount $res_{available}$ of available computational resources as a global variable. This variable has to be checked before carrying out any new contingency analysis and updated after each contingency analysis (exactly as if the contingency analyses were performed sequentially), in order to make sure that the available computational budget is respected.

Problem definition: a Euclidean contingency space $\mathscr{X}$, an objective function $O : \mathscr{X} \to \mathbb{R}$ and a threshold $\gamma \in \mathbb{R}$.

Algorithm parameters: the parameters $\lambda_0 = [\mu_0, \Sigma_0]$ of the initial Gaussian sampling distribution, the size s of the sample drawn at each iteration, the number m of best solutions chosen at each iteration and the maximal number i_{max} of iterations to be done within one "sub-run" of the algorithm.

Input: the amount $res_{available}$ of available computational resources.
Output: a set $\mathscr{X}_{dang}$ of elements of $\mathscr{X}$ such that $O(x) \geq \gamma$.
Algorithm:

Step 1. Set $i = 0$ and set $\mathscr{X}_{dang}$ to the empty set.

Step 2. Set S_i, P_i and S'_i to empty sets.

Step 3. Draw independently s elements of $\mathscr{X}$ according to the probability distribution $Gauss_{\mathscr{X}}(\cdot, \lambda_i)$ and store them in S_i.

Step 4. For every element $x \in S_i$:
If $res_{available} > 0$, compute $o = O(x)$, set $res_{available} = res_{available} - 1$ and add the pair (x, o) to P_i .

Step 5. Identify in P_i the pairs for which $o \geq \gamma$ and set their x values in $\mathscr{X}_{dang}$ if they are not already in it.

Step 6. If $res_{available} = 0$, go to **Step 10.** Else, go to **Step 7.**

Step 7. Identify in P_i the m pairs with the highest values of o and set their x values in S'_i.

Step 8. If $i < i_{max} - 1$, set $\mu_{i+1}[j] = \frac{1}{m} \sum_{x \in S'_i} x[j]$ for $j = 1, \ldots, n$,

$\Sigma_{i+1} = \frac{1}{m} \sum_{x \in S'_i} (x - \mu_{i+1})(x - \mu_{i+1})^T$, $\lambda_{i+1} = [\mu_{i+1}, \Sigma_{i+1}]$, set $i \leftarrow i + 1$ and go to **Step 2.**
Else, go to **Step 9.**

Step 9. Set $i = 0$, set S_i, P_i and S'_i to empty sets and go to **Step 3.**

Step 10. Output $\mathscr{X}_{dang}$ and stop.

Figure 2.10: A comprehensive iterative sampling algorithm for identifying the elements such that a function $O : \mathscr{X} \to \mathbb{R}$ exceeds a threshold γ when the space $\mathscr{X}$ is Euclidean, while respecting a computational budget $res_{available}$.

2.4.3 Multimodal objective function

Repeating the basic iterative sampling algorithm several times is not only a way to take better advantage of the available computational resources, but also a solution to deal with the potential multimodality of the objective function.

Depending on the security assessment problem at hand, it is indeed probable that the objective function has several local maxima, and, subsequently, that some dangerous contingencies (i.e., some points such that $O(x) \geq \gamma$) are located in different areas of the search space, around some of these maxima. Note that the algorithm does not know a priori the number of maxima of the objective function.

As the basic iterative sampling algorithm uses simple Gaussian laws as sampling distributions, one run of this algorithm will converge towards one of these maxima. It usually identifies during its execution some dangerous contingencies located in the neighborhood of this maximum and possibly a few dangerous contingencies located in other areas encountered over the iterations (but neither numerous nor severe enough with respect to the other dangerous contingencies found to direct the sampling process towards their location). With such an algorithm, there is a risk of missing some dangerous contingencies corresponding to the other maxima of the objective function if this function is multimodal.

Different frameworks can be found in the literature to adapt the cross entropy method, from which our basic iterative sampling algorithm for dangerous contingency identification is inspired, to the case where the objective function is multimodal. It is for instance possible to use the Fully Adaptive Cross-Entropy (FACE) algorithm introduced in [23], or to use mixture distributions instead of Gaussian laws in the cross-entropy algorithm as proposed in [24].

The strategy of re-launching the basic iterative sampling algorithm as long as the available computational resources have not been exhausted also allows to avoid missing a maximum of a multimodal objective function. Thanks to the stochastic aspect of the basic iterative sampling algorithm, we can expect that, when it is executed several times, it converges towards some different local maxima. Even if one of the "sub-runs" of the comprehensive iterative sampling approach converges towards an extremum that has already been reached by one or several of the previous "sub-runs", we also hope that this will allow to identify more dangerous contingencies located in this area (if there are still dangerous contingencies in this area that have not been identified yet). Repeating the basic iterative sampling algorithm several times thus helps increasing the number of dangerous contingencies identified, whatever the region of the search space each "sub-run" converges to.

2.5 The objective function

This section reminds the role of the objective function in the developed iterative sampling algorithm and proposes several ways to define it in a general context, independently from the considered search space (which can be either a Euclidean space as in the problem addressed in this chapter or a discrete space as in the problem addressed later in Chapter 3).

2.5.1 Role and definition

The objective function O is a function defined over the contingency space, taking values larger than the threshold γ for the dangerous contingencies. This function as well as the threshold γ are chosen beforehand by the user, based among others on how dangerous contingencies are defined. The value $O(x)$ of the objective function for a given contingency x is referred to as its severity.

The function O is used at every iteration of our iterative sampling algorithm to select the most severe contingencies in the current sample and direct the next sampling distribution towards them, based on the assumption that contingencies with similar severities are located close to each other in the search space. Note that, to do so, the algorithm does not exploit the values of the objective function for themselves but rather the order relation between the severities of the contingencies in each sample. As a consequence, the performances of our iterative sampling framework are invariant with respect to any monotonic transformation applied to the chosen objective function (the value of γ being adjusted accordingly).

We propose to define the value of the objective function for a given contingency based on the results of the simulation of the occurrence of this contingency with the available algorithmic tool. The different simulation tools that can be used have been presented in the introduction. Sorted by growing algorithmic complexity, they include DC power flow, AC power flow, voltage stability analysis and transient stability analysis. The system operator usually chooses among them the one corresponding to the complexity of the studied phenomena (as an example, in the first case study presented in Chapter 4, AC load-flows are executed to evaluate the severity of $N-3$ contingencies). From there, he can build an objective function relying on either global or equipment-based criteria.

2.5.2 Global criteria

When assessing the security of a power system, it is very useful to have global indicators of the security level of the system, which allows quick diagnosis and decision

making. It comes naturally to use such a global criterion to evaluate the severity of a contingency in the problem addressed in this thesis. Some detailed examples of ways to define the objective function are provided below.

2.5.2.1 Example 1: impact of unsupplied energy

A natural approach used by the TSOs to quantify the severity of a contingency is to evaluate its impact on unsupplied energy. To do so, the value of load disconnections may be estimated using customer damage functions (see [25] and [26]). Another way of costing load interruptions is to use the Value of Lost Load (VOLL), defined as the average value that customers attach to the loss of one kilowatt for one hour (see [27]). In order to reflect the impact of unsupplied energy, the value of the objective function can be computed in a straightforward way by multiplying the value of the amount of load disconnected by the time required to resynchronize and load (or directly replace) the lost units. If a contingency does not result in any load loss, its value of the objective function is simply set equal to 0. Depending on the security assessment problem addressed by the user, the threshold γ can be set equal to a value slightly superior to 0 or to a larger value in order to define the dangerous contingencies.

2.5.2.2 Example 2: distance of the system variables to their limits

The objective function can be built by evaluating the distance between the values of the system variables, such as bus voltages or line flows, and their operating limits. For instance, when focusing on line flows, such an approach would first require to check for each line in the system if the maximal flow the line is admitted to carry is reached, and, if it is the case, to subtract this limit from the flow in the line so as to quantify the limit violation. These values would then have to be summed or averaged so as to obtain a unique value that could be used as objective function.

2.5.2.3 Example 3: voltage stability indices

Many methods have been proposed in the literature to determine indices of voltage stability based on the distance of the post-contingency state to voltage collapse. Some of them are listed in [28] and [4]. Such indices are either based on load-flow feasibility, like the L indicator introduced in [29], or on a steady-state stability analysis, like the smallest singular value of the Jacobian matrix of the power flow equations (see [30]).

In order to apply our iterative sampling approach for performing a voltage stability study, the objective function can easily be derived from one of these indices.

2.5.2.4 Example 4: exploiting algorithmic properties of the simulation tools

We investigated in this work a way to take advantage of the algorithmic properties of the simulation tools used in the analysis part of the security assessment task. In particular, we chose to relate the value of the objective function to the number of iterations that are needed by an AC load-flow to converge when run in the post-contingency situation. It is indeed likely that the closer an event is from instability, the more the load-flow algorithm has to iterate to reach convergence.

This is illustrated by Figure 2.11, depicting the relationship between the global system load and the number of iterations needed by the AC load-flow computation using Newton's algorithm. If for some contingencies the load-flow diverges or has not converged when the maximum number of iterations (equal to 10 here) has been reached, the value of the objective function for these contingencies is set equal to this maximum number of iteration plus one. This ensures that the function O takes its maximum values on the dangerous contingencies.

This definition of the objective function has been used in one of the case studies reported in Chapter 4 (concerning an $N-3$ security analysis on IEEE 118 bus test system).

2.5.3 Equipment-based criteria

As proposed in the previous section, using global criteria to define an objective function required by the approach is the most intuitive way to proceed. It provides the user with a synthetic vision of the security level of the system and of the contingencies that would degrade it, which facilitates efficient decision making.

However, in some cases, the user may want to have a precise insight on the operating conditions of some specific equipments playing a key role in the security of the system, like long transmission corridors, cross-border lines, or older system elements that were not dimensioned for today's requirements, whose loss might initiate cascades that would quickly propagate to the rest of the system, such a situation becoming difficult to mitigate. TSOs may then want to be able to identify the contingencies that would specifically affect these critical equipments.

Moreover, the computation of a global severity index introduces a risk of masking some dangerous situations. With a global criterion based on state variables of the equipments of the system as the one introduced in Subsection 2.5.2.2, it is indeed possible that a contingency leading to a small increase in the level of stress of several equipments has a higher value of the objective function than a contingency resulting in a higher degradation of the state of one single equipment. The system might be closer to instability after the occurrence of this latter contingency, whereas the objective function

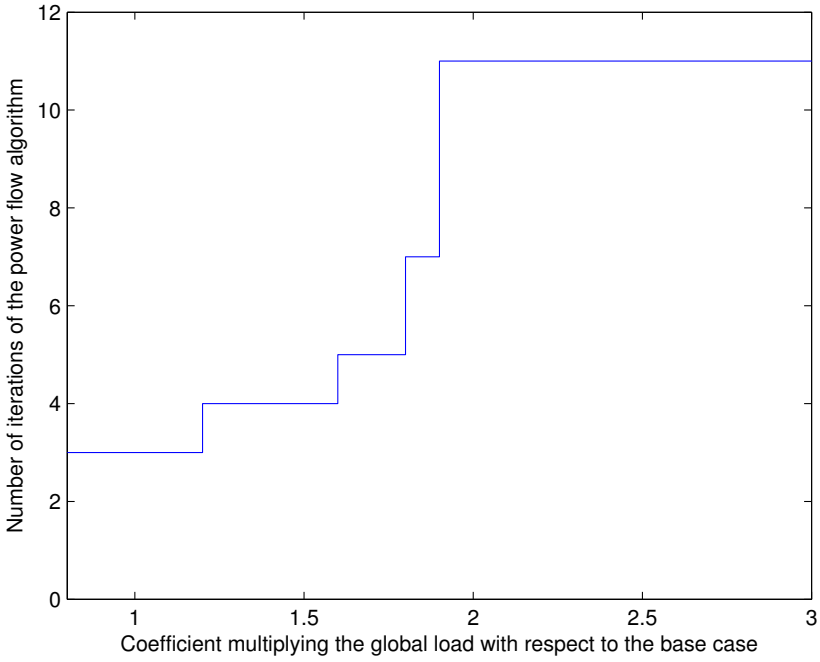


Figure 2.11: Evolution of the number of iterations to convergence of the AC load-flow algorithm when the load increases monotonically. The considered test system is the IEEE 118 bus system.

would not reflect this.

All these reasons suggest that it may also be relevant to adopt an equipment-based objective function. It can for instance reflect the post-contingency operating conditions (e.g., line flows or bus voltages) of one particular equipment identified by the user as a weak-point of the system, or of a small set of such equipments.

2.5.3.1 Example 1: nodal voltage collapse proximity indicator

When performing a voltage stability analysis, the objective function can be defined by focusing on one particular load bus and computing its voltage collapse proximity in-

dicator (see [31]). The value of this indicator is based on the change of the reactive consumption of the considered load bus and on the change of the total reactive generation in the system after the occurrence of the considered contingency. For the i^{th} load bus of the considered system, it can be computed using the following equation:

$$VCPI_i = \frac{\sum_{j \in \Omega_G} \Delta Q_{G_j}}{\Delta Q_i} \tag{2.1}$$

where Ω_G is the set of generator buses in the system, ΔQ_{G_j} is the change of reactive generation of the j^{th} generator and ΔQ_i is the change of reactive consumption at the i^{th} load bus.

It has been shown in [31] that this indicator takes values comprised between 5 and 10 when the state of the system is far from voltage collapse, and takes significantly higher values (tending to infinity) when the system is approaching voltage collapse. The threshold γ above which a contingency is considered as dangerous could thus be set equal to 10.

2.5.3.2 Example 2: post-contingency line flows

In two of the case studies proposed in Chapter 4, we focus on a single weak transmission line and allocate to the objective function the value of the loading rate induced by a contingency on the considered line. A contingency is thus considered as dangerous if this value exceeds the threshold defined by the operator according to the physical properties of the line.

2.6 Illustration on a simple power system security assessment problem

We now propose a preliminary illustration of the proposed iterative sampling algorithm on a small and simple dangerous contingency identification problem.

2.6.1 Setting

We have chosen to run this first experiment on the IEEE 30 bus system (described in [32]). The problem addressed here, which was also studied in [33], is the identification of dangerous contingencies for static voltage security, and more specifically for static loadability.

In this setting, a *contingency* is defined as being an homothetic increase or decrease of the load with respect to the base case, modeled by a coefficient $x \in [0.25; 2.5]$ multiplying the total load in the base case. The contingency space considered here is made up of a set of configurations of this system characterized by their respective load patterns. These different configurations are obtained by multiplying the load in the base case by a real coefficient x in $[0.25; 2.5]$. Each contingency is modeled by the corresponding value of x, and the contingency space considered in this problem is thus the interval $[0.25; 2.5]$.

The *objective function* chosen in this illustration is based on the algorithmic behavior of an Optimal Power Flow (OPF) solver. Here, the OPF algorithm is used to optimize the generation dispatch and the generator voltages to minimize generation cost (see [34]). To evaluate the objective function for a contingency x, an OPF algorithm is run on the configuration of the system obtained when multiplying the global load in the base case by x. If the algorithm converges towards a feasible solution, the value of the objective function is set equal to the number of iterations required to reach this solution. In all other cases, the value of the objective function is set equal to an arbitrary large value (1000), chosen greater than the maximal number of iterations of the algorithm. The choice to define the objective function this way is based on the assumption that, the higher the number of iterations, the more the system is stressed and the closer it operates from its stability limits – which are considered to be reached when the algorithm does not converge.

We thus define a contingency as non-dangerous if the corresponding demand level can be served by the available active and reactive generation capacity while respecting voltage constraints (i.e., if the OPF problem corresponding to this variation of the load is feasible). This contingency is considered as *dangerous* if it is not the case (i.e., if the OPF algorithm does not converge or diverges). The maximal number of iterations performed by the algorithm being equal to 500, an appropriate value for γ is any value in $]500; 1000]$.

For the sake of simplicity, we chose to apply the iterative sampling algorithm that was provided in Figure 2.2, which does not take into account any constraint on the available computational resources. The algorithm parameters used in these simulations are the following: $\lambda_0 = [1, 1]$, $s = 20$, $m = 5$ and $i_{max} = 7$.

2.6.2 Results

When applied to this problem, our iterative sampling algorithm outputs a set of real values for x corresponding to dangerous contingencies.

Figure 2.12 illustrates the behavior of the algorithm during a typical run. At the

end of each iteration, we have computed the ratio between of dangerous contingencies contained in the current sample and the sample size. The corresponding values are represented in dark blues on the figure. As a comparison, we have also drawn iteratively 7 samples of 20 points from the contingency space using a classical Monte Carlo sampling process, and also computed the rate of dangerous contingencies in each of these samples, represented in light blue on the figure.

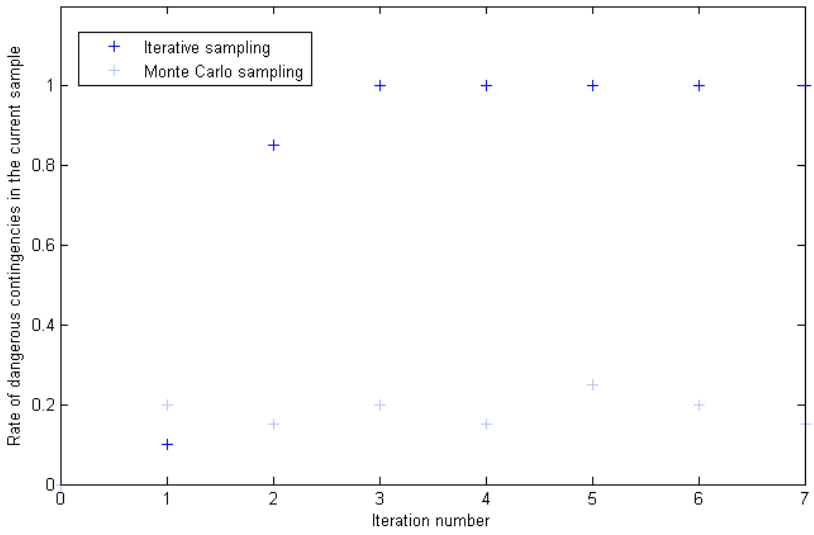


Figure 2.12: Evolution of the rate of dangerous contingencies contained in the successive sample drawn from the contingency space with our iterative sampling approach and with a classical Monte Carlo sampling.

We observe that the rate of dangerous contingencies in the current sample rapidly grows from 0.1 at the first iteration to 0.85 at the second iteration and 1 at the third one with our iterative sampling method, which significantly outperforms the classical Monte Carlo sampling that only comes across on average 18 % of dangerous contingencies in each of the samples drawn from the contingency space (which is also the rate of dangerous contingencies within the contingency space). These results are due to the fact that, at the end of each iteration, the mean and variance of the normal sampling distribution to be used in the following iteration are adjusted according to the 5 more

severe contingencies contained in the current sample, which are almost all dangerous at iterations 1 and 2 and then all dangerous at the following iterations. Contrary to the iterative sampling approach, the Monte Carlo sampling process does not exploit the properties of the previously drawn samples when drawing the following one, and is therefore not able to yield samples whose rate of dangerous contingencies exceeds the one of the entire contingency space.

Note: in practice, a possible engineering approach that could be adopted to solve such a problem would be to assume that there is a threshold on the values of x above which all the contingencies are dangerous and under which all the contingencies are non-dangerous. The value of this threshold could then be estimated thanks to a dichotomy approach. By using such an approach, we found out that this threshold is equal to 2.1. This assumption is coherent with the results of the iterative sampling algorithm, since neither a dangerous contingency with a value of x under this threshold nor a non-dangerous contingency with a value of x above it has been encountered. Note that, if the objective function was rather oscillating several times around one such threshold, our iterative sampling approach would be able to identify dangerous contingencies corresponding to these oscillations.

3

Embedding the contingency space in a Euclidean space

While the iterative sampling approach described in Chapter 2 was meant to be applied to a Euclidean search space, the contingency spaces met in practice are rather discrete. In order to apply an iterative sampling algorithm in such a context, we propose in this chapter to embed the contingency space in a Euclidean space, over which continuous sampling distributions can be defined.

3.1 Introduction

In most dangerous contingency identification problems, the studied contingency space is discrete. It is obvious when performing an $N-k$ security analysis. It is also the case when a discrete number of different generation and consumption patterns likely to occur (due for instance to the uncertainties about the amount of power produced with renewable sources) is used as contingency space.

The iterative sampling algorithm that was introduced in Figure 2.10 cannot be applied as such to a discrete contingency space since it requires the existence of a metric and works with sampling distributions that are defined on a Euclidean space. It is therefore necessary to embed the contingency space in a Euclidean space, denoted by $\mathscr{Y}$ in the following. A simple and intuitive choice for such an embedding space is for instance the plane, $\mathbb{R}^2$, and $\mathbb{R}^n$ $(n > 2)$ can also be adopted in more complex problems.

Such an embedding procedure requires to define functions allowing to navigate between the original contingency space and its Euclidean embedding space, and conversely.

3.1.1 Projection operator

The first parts of this embedding process is the definition of a projection operator on the contingency space, denoted by $Proj : \mathscr{X} \rightarrow \mathscr{Y}$. This operator calculates for each contingency a point in the chosen metric space ideally in such a way that contingencies which are projected on nearby points have similar values of the objective function.

As an example, a contingency consisting of one single equipment outage can be projected in $\mathbb{R}^2$ as a point whose coordinates represent the location of this equipment in the geographical map of the system.

3.1.2 Pre-image function

The second part of the embedding process consists in a reverse mapping (that we also name pre-image computation) of the embedding space, so as to associate to each point of this latter space an element of the original discrete contingency space. We name the function fulfilling this role the pre-image function, and denote it by $PreImage : \mathscr{Y} \rightarrow \mathscr{X}$ in the following. In the case where the contingency space is discrete, our iterative sampling algorithm will be run on the embedding space $\mathscr{Y}$. In order to evaluate the severity of the points drawn from this space by the algorithm, the objective function (defined on the space $\mathscr{X}$) will have to be composed with the pre-image function.

We propose to define this pre-image function in a very simple way, by associating to each point of the Euclidean embedding space the contingency from which it stands

the closest according to the chosen metric. This procedure partitions the embedding space into the Voronoi diagram of the set of points associated to the contingencies, each surface of this diagram corresponding to a different contingency.

3.1.3 Whole metrization process

What we will refer to as the metrization process in the following includes the three following steps: choosing an adapted Euclidean space, associating to each contingency a point of this space and defining the pre-image function that allocates to each point of this space an element of the original contingency set. Figure 3.1 gives a schematic view of the relationships between the two different spaces involved in this process.

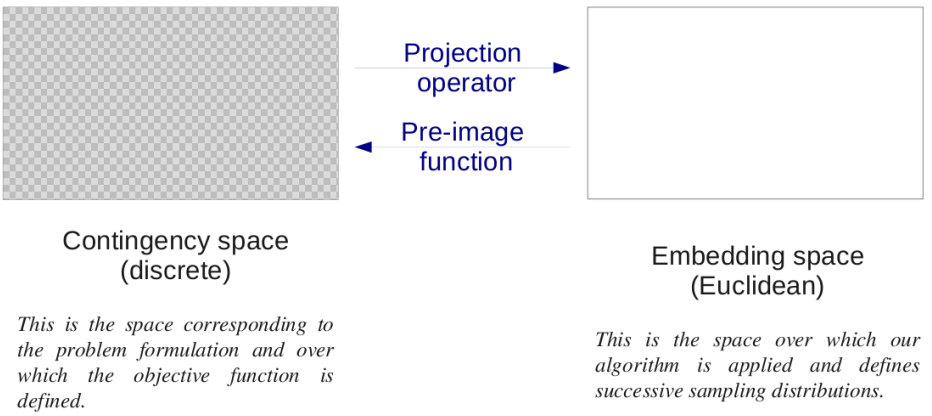


Figure 3.1: Schematic representation of the embedding of a discrete contingency space in a Euclidean space.

Two detailed examples of metrization procedures will be presented in the following sections to embed the set of all $N-k$ line tripping contingencies in R^{2k}.

3.2 Example 1: embedding the space of all $N-k$ line tripping contingencies in $\mathbb{R}^{2k}$ by exploiting the geographical coordinates of the buses

In order to explain the strategy we have chosen to embed this specific contingency space in $\mathbb{R}^{2k}$, we will first show how to embed the set of all $N-1$ line tripping contingencies in the plane ($\mathbb{R}^2$).

To define the projection operator between this contingency space and the plane, we propose to represent each $N-1$ line tripping contingency in $\mathbb{R}^2$ by the midpoint of the segment modeling the lost branch in the two-dimensional map of the system. This leads to an easily interpretable representation since the user is familiar with the geographical position of the branches in the studied system.

As for the pre-image function, it is defined as proposed in the previous section: we associate to each point of the plane the contingency whose projection it stands the closest to. This implicitely defines the Voronoi diagram of the set of projected contingencies, and every point of a Voronoi cell is associated by the pre-image function to the same contingency (the contingency whose projection is the center of the cell).

The rationale behind this procedure is based on the assumption that, if two points are close to each other in the plane, the contingencies they are associated with by the pre-image function are likely to have similar effects on the post-contingency steady-state. In the case treated here of an $N-1$ study limited to branch outages, we thus expect that two neighboring points of the plane will be associated to the tripping of two branches of the system that are geographically close to each other, and that the effect of these two outages on the security of the system are similar in some sense.

Figure 3.3 illustrates this approach on the IEEE 14 bus test system, whose classical one-line diagram is reminded on Figure 3.2 to enable comparison. The red crosses on this figure correspond to the projection of the $N-1$ contingencies in the plane (as the midpoint of the lost line). The black lines delimit the zones of the plane associated to each contingency by the pre-image process, i.e. the Voronoi cells of the red crosses representing the contingencies in $\mathbb{R}^2$.

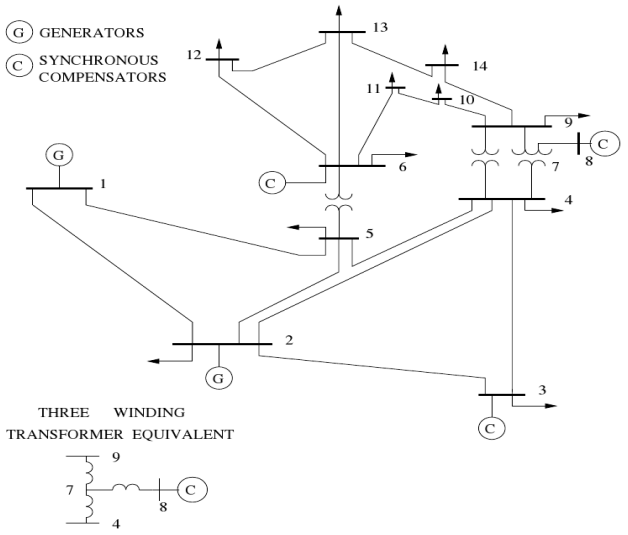


Figure 3.2: One-line diagram of the IEEE 14 bus test system.

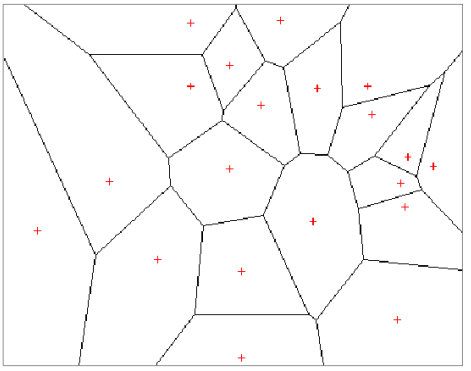


Figure 3.3: Representation of the $N-1$ contingencies as the midpoints of the lost line and corresponding Voronoi diagram (IEEE 14 bus test system).

The above embedding procedure may be easily extended to embed $N-k$ $(k>1)$ line tripping contingencies in $\mathbb{R}^{2k}$.

The projection of an $N-k$ line tripping contingency, denoted by the k-tuple $(l_1, l_2, \ldots, l_k)$ where every l_i refers to a transmission line, in $\mathbb{R}^{2k}$ is defined as the $2k$-tuple $(y_1, y_2, \ldots, y_{2k})$, whose k^{th} pair of components corresponds to the coordinates of the midpoint of line l_k in the geographical (two-dimensional) representation of the system.

Once such a projection operator between the contingency space and $\mathbb{R}^{2k}$ has been built, the pre-image $(l_1, l_2, \ldots, l_k)$ of a point y of the Euclidean space $\mathbb{R}^{2k}$ with coordinates $(y_1, y_2, \ldots, y_{2k})$ might be computed as follows. To identify the component l_1, we take the two first components of the $2k$-dimensional vector and consider them as the coordinates of a point in the plane. From there, we identify the first element of the tuple $(l_1, l_2, \ldots, l_k)$ as if we were dealing with an $N-1$ contingency: we use the projection of the contingencies in the plane as the midpoints of the corresponding lost branches and search among them for the one standing the closest to the point of coordinates (y_1, y_2). By taking the two second components of the $2k$-dimensional vector, we identify l_2 using the same procedure, and then similarly $(l_3, \ldots, l_k)$.

In order to make sure that the obtained k-tuple is made of distinct branches, we have slightly modified the pre-image computation procedure as follows. First, we consider that the elements of a k-tuple are identified sequentially. At every step j, we check after having identified l_j whether there exists in $\mathscr{X}$ a k-tuple whose first j elements are $(l_1, l_2, \ldots, l_j)$, whatever their order. If it is not the case, we choose as l_j the second closest branch to the considered point of the Euclidean space. There is again a similar checking on this new l_j and the procedure repeats if necessary.

Note: this procedure for computing the pre-image of a point of the Euclidean embedding space does not necessary yield the point of $\mathscr{X}$ whose projection is the nearest neighbor the considered point of $\mathscr{Y}$. However, the greedy nature of the pre-image computation is essential for keeping the computational complexity of the algorithm linear with respect to the number of contingency analyses (in accordance with the hypothesis formulated in Chapter 1.2.2.3) in spite of the combinatorial structure of $\mathscr{X}$.

3.3 Example 2: embedding the space of all $N-k$ line tripping contingencies in $\mathbb{R}^{2k}$ by computing "electrical" coordinates of the buses

3.3.1 An alternative to the geographical bus coordinates: "electrical" coordinates

When projecting $N-k$ line tripping contingencies as points of $\mathbb{R}^{2k}$, we expect that two neighboring points of $\mathbb{R}^{2k}$ will be associated to two $N-k$ contingencies with similar effects on the security of the system. However, one main drawback of the strategy based on the geographic coordinates of the midpoints of the branches is that this latter assumption might not always be justified. Indeed, the geographical distance between two equipments may be very poorly correlated with their electrical distance, that seems to be a more natural distance. For instance, the loss of two different neighboring lines might have very different consequences for the security of the system, and conversely some outages on two lines that are geographically remote from each other might have similar effects if these lines are not that distant from an electrical point of view.

To circumvent this problem while still having the possibility to define a projection operator based on the location of the midpoints of the lines in a two-dimensional space, we suggest to represent the buses (and subsequently the midpoints of the segments modeling the transmission lines) differently on the plane: rather than positioning them according to their geographical location, we propose to represent them in a way such that the distance between every pair of buses is correlated to their electrical distance.

A relevant way to define such an electrical distance between two buses is to compute the reduced impedance of the network between them. This is done by reducing the admittance matrix of the system to these two buses and by computing the modulus of the inverse of this value.

Assuming that a distance matrix collecting the inter-bus electrical distances computed as proposed above is available, we now propose an algorithm for computing two-dimensional bus coordinates (independent from the geographical ones) such that the distances between every pair of buses in this representation reflect these electrical distances. This approach is based on an optimization algorithm, borrowed from the multidimensional scaling (MDS) literature (see [35]), known as the SMACOF algorithm (see [36]). The acronym SMACOF stands for Scaling by Majorizing a Complicated Function. This algorithm aims at minimizing the sum of the squared differences between the Euclidean distances between the computed "electrical" coordinates and the given electrical distances.

If we denote by $D \in \mathbb{R}^{n\times n}$ the matrix containing the inter-bus electrical distances,

by d_{ij} the element of this matrix corresponding to the electrical distance between buses i and j and by $X \in \mathbb{R}^{n\times 2}$ the vector of "electrical" bus coordinates we want to compute, the problem solved by this algorithm can thus be written as:

$$\arg\min_{X} f(X) \,, \tag{3.1}$$

where

$$f(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\sqrt{\sum_{k=1}^{2} (x_{ik} - x_{jk})^2} - d_{ij} \right)^2 . \tag{3.2}$$

The function f defined in Equation (3.2) can be expanded as follows:

$$\begin{aligned} f(X) = & \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=1}^{2} (x_{ik} - x_{jk})^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(d_{ij}\right)^2 \\ & - 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\sqrt{\sum_{k=1}^{2} (x_{ik} - x_{jk})^2} \right) d_{ij} \,. \end{aligned} \tag{3.3}$$

The first term of this sum can also be written:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=1}^{2} (x_{ik} - x_{jk})^2 = tr(X'AX) \,, \tag{3.4}$$

with $A \in \mathbb{R}^{n\times n}$ being such that $a_{ii} = n-1$ and $a_{ij} = a_{ji} = -1$.

The second term of $f(X)$ does not depend on X and can be seen as a constant, so we set:

$$k_0 = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(d_{ij}\right)^2 . \tag{3.5}$$

In the third term of $f(X)$, we denote by $dist_{i,j}(X)$ the Euclidean distance between buses i and j:

$$\sqrt{\sum_{k=1}^{2} (x_{ik} - x_{jk})^2} = dist_{i,j}(X) \,. \tag{3.6}$$

Given (3.4), (3.5) and (3.6), Equation (3.3) can be written concisely as:

$$f(X) = tr(X'AX) + k_0 - 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} dist_{i,j}(X)\, d_{ij} \,. \tag{3.7}$$

The third term of this expression is non-convex and makes the resolution of the problem (3.1) difficult. To address this problem, it is possible to majorize this term by a convex expression to get a new objective function, easier to minimize. The SMACOF algorithm exploits the following majorization, based on the Cauchy-Schwartz inequality:

$$\begin{aligned}\sum_{k=1}^{2}(x_{ik}-x_{jk})(y_{ik}-y_{jk}) &\leq \Big(\sum_{k=1}^{2}(x_{ik}-x_{jk})^2\Big)^{1/2} \\ &\quad \times\Big(\sum_{k=1}^{2}(y_{ik}-y_{jk})^2\Big)^{1/2} \\ &\leq dist_{ij}(X)dist_{ij}(Y)\,,\end{aligned} \tag{3.8}$$

where $Y \in \mathbb{R}^{n\times 2}$ can be interpreted as another set of coordinates for the buses.

If we multiply both sides of the inequality by (-1) and divide by $dist_{ij}(Y)$, we obtain:

$$-dist_{ij}(X) \leq \frac{\sum_{k=1}^{2}(x_{ik}-x_{jk})(y_{ik}-y_{jk})}{dist_{ij}(Y)}\,. \tag{3.9}$$

By summing over $i = 1\ldots n$ and $j = i+1\ldots n$ we obtain the majorizing expression:

$$\begin{aligned}-2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} dist_{i,j}(X)\, d_{ij} &\leq -2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=1}^{2}\frac{d_{ij}}{dist_{ij}(Y)} \\ &\quad \times(x_{ik}-x_{jk})(y_{ik}-y_{jk}) \\ &\leq -2tr(X'B(Y)Y)\,,\end{aligned} \tag{3.10}$$

with $B(Y) \in \mathbb{R}^{n\times n}$ being such that:

$$\begin{aligned} b_{ij} &= \begin{cases} -\dfrac{d_{ij}}{dist_{ij}(Y)} & \text{for } i \neq j \text{ and } dist_{ij}(Y) \neq 0 \\ 0 & \text{for } i \neq j \text{ and } dist_{ij}(Y) = 0 \end{cases} \\ b_{ii} &= -\sum_{j=1, j\neq i}^{n} b_{ij}\,. \end{aligned} \tag{3.11}$$

By combining (3.7) and (3.10), the function f itself can be majorized:

$$f(X) \leq tr(X'AX) + k_0 - 2tr(X'B(Y)Y) = g(X) \; . \tag{3.12}$$

The function g is a quadratic function of X. The minimum of the function g is obtained when its derivative is equal to zero, i.e.:

$$\nabla g(X) = 2AX - 2B(Y)Y = 0 \; . \tag{3.13}$$

The value of X minimizing $g(X)$ is such that:

$$AX = B(Y)Y \; . \tag{3.14}$$

As the inverse A^{-1} does not exist since A is not full rank, this linear equation in X cannot be solved by premultiplying both sides of (3.14) by A^{-1}. The Moore-Penrose inverse, given by $A^{+} = (A + \mathbf{1}_{n,n})^{-1} - n^{-2}\,\mathbf{1}_{n,n}$ (where $\mathbf{1}_{n,n}$ is the matrix such that $\mathbf{1}_{n,n}(i,j) = 1 \; \forall (i,j) \in \{1,\ldots,n\}^2$), is used in the SMACOF algorithm. The matrix X minimizing $g(X)$, and subsequently $f(X)$, is the following:

$$X = A^{+}B(Y)Y \; . \tag{3.15}$$

It can be shown that the solution computed from (3.15) is such that $f(X) \leq f(Y)$. The SMACOF algorithm exploits this property to iteratively compute solutions with decreasing values of f. The solution computed at iteration i, denoted by X_i, is equal to $A^{+}B(X_{i-1})X_{i-1}$. The tabular version of the procedure used in our simulations is given in Figure 3.4.

Problem definition: an n-by-n distance matrix for the set of n power system buses considered.
Algorithm parameters: a small positive value ε, which is the minimum decrease of f after an iteration for not stopping the iterative process, and a maximal number of iterations i_{max}.

Output: a matrix $X^{elec} \in \mathbb{R}^{n\times 2}$ of "electrical" bus coordinates.
Algorithm:

Step 1. Set X_0 to a random $n \times 2$ matrix.
Set iteration counter $i = 0$.
Step 2. Compute $f_0 = f(X_0)$. Set $f_{-1} = f_0$.
Step 3. While $i = 0$ or $\big((f_{i-1} - f_i) > \varepsilon$ and $i \leq i_{max}\big)$ do:
Set $i \leftarrow i+1$.
Compute $B(X_{i-1})$ by using Equation (3.11).
Set $X_i = A^+ B(X_{i-1}) X_{i-1}$.
Compute $f_i = f(X_i)$.
Set $X_{i+1} = X_i$.
Step 4. Set $X^{elec} = X_i$. Output X^{elec}.

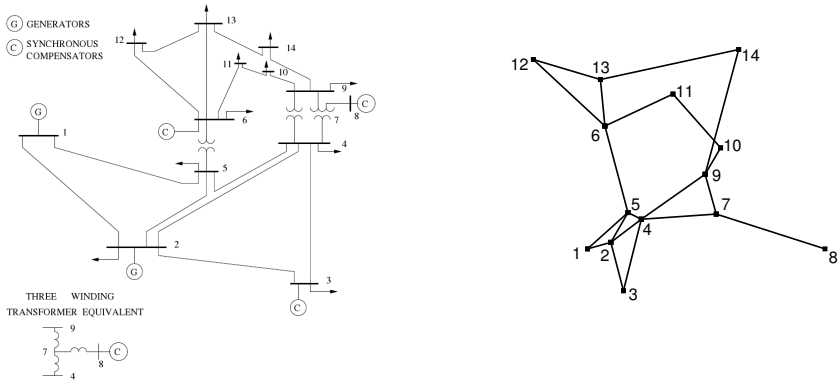
Figure 3.4: A tabular version of the SMACOF algorithm for computing two-dimensional coordinates of a set of n buses based on their electrical distances.

3.3.2 Illustration: representation of the buses of the IEEE 14 bus test system according to their electrical distances

We illustrate the procedure described previously on the IEEE 14 bus test system. The one-line diagram of this network is reminded on Figure 3.5a. Figure 3.5b represents on the plane the position of the buses of the IEEE 14 bus system as outputted by the algorithm 3.4, as well as the lines connecting these buses.

Obviously, the positions of the buses on the classical one-line diagram do not give a good image of their electrical distances. For example, nodes 1, 2, 4 and 5 are much closer from an electric point of view than they appear on the original one-line diagram. It is also worth noticing that node 8, which is connected to nodes 4 and 9 through a three windings transformer, appears quite remote on the one-line diagram drawn on Figure 3.5b while it is not the case on the other diagram. This was expected since the windings of a transformer generally have a reactance whose value is in the range of the

reactance value of a few tens of kilometers long transmission line.



(a) The "classical" one-line diagram

(b) One-line diagram that positions buses according to their electrical distances

Figure 3.5: Representation of two one-line diagrams of the IEEE 14 bus test system. Figure (a) gives the classical one-line diagram, as published in the literature. The diagram represented on Figure (b) positions the nodes according to their electrical distances by using the algorithm provided in Figure 3.4.

3.3.3 Using such an "electrical" representation to embed the contingency space in a Euclidean space

Once the studied network has been represented according to the electrical inter-bus distances, we propose to adopt a procedure similar to the one presented in the previous section for embedding the contingency space in a Euclidean space.

In the case of an $N-1$ analysis, the contingencies can be projected in the plane as the midpoints of the segments representing the corresponding lost line in the "electrical map" (the map built from the electrical distances). The pre-image function associates to each point of $\mathbb{R}^2$ drawn by the iterative sampling algorithm the contingency whose projection it is the closest to.

Figure 3.6 shows the projection of the $N-1$ contingencies in the plane when working with the one-line diagram of IEEE 14 bus test system introduced on Figure 3.5b (i.e., a representation of this system based on the electrical distances between buses). As in Section 3.2, each $N-1$ contingency is projected on the plane as the midpoint

of the tripped transmission line. The sub-areas of the plane associated to each of these contingencies by the pre-image process are also represented.

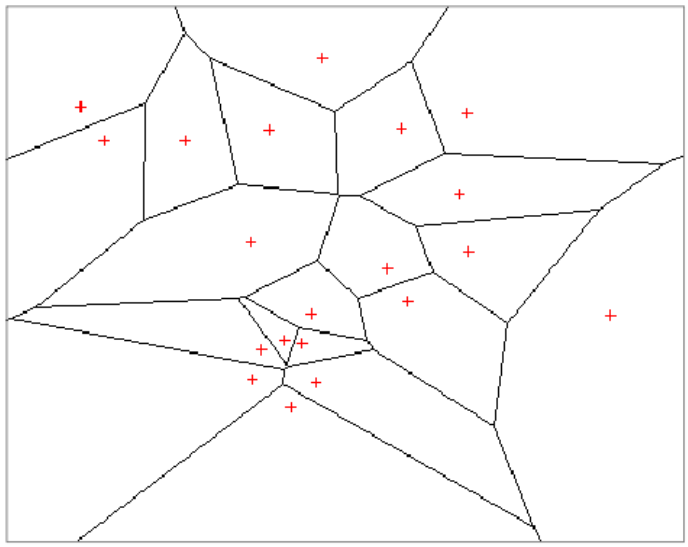


Figure 3.6: Representation of the $N-1$ contingencies as the midpoints of the lost lines and of the area of the plane associated to each of them (IEEE 14 bus test system, while considering here that the buses are no longer positioned according to their geographical coordinates but rather to their electrical distances).

This embedding process can be extended exactly like in the previous section when performing an $N-k$ analysis when $k > 1$.

3.4 Updated version of our basic and comprehensive iterative sampling algorithms

When searching for dangerous contingencies in a discrete contingency space, the same basic and comprehensive iterative sampling approach as the ones proposed in Chapter

2.4 can be applied once this space has been embedded in a Euclidean space.

A fully specified version of our basic iterative sampling algorithm adapted to discrete search spaces is provided in Figure 3.7. From an algorithmic point of view, the differences between this algorithm and the one that was presented in Figure 2.2 are listed hereafter:

- the definition of the problem to which this algorithm is applied now includes the pre-image function in addition to the contingency space $\mathscr{X}$, the objective function O and the value $\gamma \in \mathbb{R}$ defining a contingency x as dangerous if $O(x) > \gamma$;
- the space in which the successive samplings are performed is now the Euclidean embedding space;
- the probability distributions the algorithm works with are defined on the Euclidean embedding space;
- it is no longer the objective function as such that is applied to points drawn from this space at each iteration, but rather its composition with the pre-image function, $O \circ PreImage$ (that is defined over the Euclidean embedding space);
- in order to save computational effort, the results of the evaluations of the objective function performed throughout the execution of the algorithm are stored in a set denoted by P under the form of pairs (x,o), where x is a contingency and $o = O(x)$ is the corresponding value of the objective function. These results can thus be reused each time the current sample drawn from the contingency space contains a contingency that has been analyzed previously;
- the information extracted from the data sampled at each step i is now stored as a set T_i of triplets (y,x,o) where y corresponds to a point drawn from the Euclidean embedding space, x to its pre-image in the contingency space and o to the value of the objective function for the contingency x;
- for the sake of clarity (and for an easier definition of the comprehensive iterative sampling algorithm based on successive calls of this basic algorithm), the basic iterative sampling algorithm now takes as input the set $\mathscr{X}_{dang}$ of dangerous contingencies that have been identified during the previous runs, the set P of pairs (x,o) formed during these runs and the amount $res_{available}$ of computational resources that remain available. It returns as output the updated versions of these three variables.
- a new stopping conditions is introduced in this fully specified algorithm: instead of stopping when the maximal number of iterations i_{max} has been reached, the

algorithms stops either when the available computational resources have been exhausted or when the pre-image of all the points of the current sample correspond to the same contingency (i.e., when all the values of x in the triplets contained in T_i are identical).

In the basic iterative sampling algorithm described in Figure 3.7, we consider that $\mathbb{R}^n$ is chosen as Euclidean embedding space for the contingency space $\mathscr{X}$. The sampling distributions the algorithm works with in this setting are n-dimensional Gaussian laws (referred to by $Gauss_{\mathbb{R}^n}(\cdot,\lambda_i)$). The parameters of these distributions are denoted by $\lambda_i = [\mu_i, \Sigma_i]$ where μ_i and Σ_i refer to the mean vector and covariance matrix of the distribution, respectively. The parameters λ_0 and Σ_0 of the initial sampling distribution are usually chosen so as to cover well the subpart of the embedding space in which the projected contingencies are located.

A fully specified version of the comprehensive iterative sampling for identifying dangerous contingencies that was proposed in Figure 2.10, adapted to the case where the contingency space is discrete and has to be embedded in a Euclidean space, is provided in Figure 3.8. Is is also built by repeating the basic iterative sampling introduced in Figure 3.7 (referred to as *BIS*) as long as the available computational resources have not been exhausted.

Problem definition: a contingency space $\mathscr{X}$, a pre-image function *PreImage* : $\mathbb{R}^n \to \mathscr{X}$, an objective function $O : \mathscr{X} \to \mathbb{R}$ and a threshold $\gamma \in \mathbb{R}$.

Algorithm parameters: the parameters $\lambda_0 = [\mu_0, \Sigma_0]$ of the initial n-dimensional Gaussian sampling distribution, the size s of the sample drawn at each iteration, the number m of best solutions chosen at each iteration.
Input: a set P collecting the pairs $(x, O(x))$ formed during previous executions of iterative sampling algorithms, the set $\mathscr{X}_{dang}$ of dangerous contingencies (i.e., points of $\mathscr{X}$ such that $O(x) \geq \gamma$) that have been identified over the previous runs and the amount $res_{initial}$ of available computational resources.
Output: the updated version of $\mathscr{X}_{dang}$ including the new dangerous contingencies identified, the updated version of P including the new pairs $(x, O(x))$ formed during this run and the amount res_{final} of remaining computational resources.
Algorithm:

Step 1. Set $i = 0$ and $res = res_{initial}$.
Step 2. Set S_i, T_i and S'_i to empty sets.
Step 3. Draw independently s elements from $\mathbb{R}^n$ according to the distribution $Gauss_{\mathbb{R}^n}(\cdot, \lambda_i)$ and store them in S_i.
Step 4. For every element $y \in S_i$:
Compute $x = PreImage(y)$.
If $x \in P$, extract from P the corresponding value of o and add the triplet (y, x, o) to T_i. Else, if $res > 0$, compute $o = O(x)$, add the pair(x, o) to P, set $res \leftarrow res - 1$ and add the triplet (y, x, o) to T_i.
Step 5. Identify in T_i the triplets for which $o \geq \gamma$ and set their x values in $\mathscr{X}_{dang}$ if they are not already in it.
Step 6. If $res = 0$, go to **Step 9.** Else, go to **Step 7.**
Step 7. Identify in T_i the m triplets with the highest values of o and set their y values in S'_i.
Step 8. If none of the stopping conditions ($res = 0$ or all values of x in T_i identical) is met, set $\mu_{i+1}[j] = \frac{1}{m} \sum_{y \in S'_i} y[j]$ for $j = 1, \ldots, n$,
$\Sigma_{i+1} = \frac{1}{m-1} \sum_{y \in S'_i} (y - \mu_{i+1})(y - \mu_{i+1})^T$ and $\lambda_{i+1} = [\mu_{i+1}, \Sigma_{i+1}]$. Set $i \leftarrow i + 1$ and go to **Step 2.**
Else, go to **Step 9.**
Step 9. Set $res_{final} = res$. Output $\mathscr{X}_{dang}$, P, res_{final} and stop.

Figure 3.7: Fully specified version of a basic iterative sampling (*BIS*) algorithm for identifying the elements such that a function $O : \mathscr{X} \to \mathbb{R}$ exceeds a threshold γ when the available computational resources are bounded, $\mathbb{R}^n$ being chosen as Euclidean embedding space.

Problem definition: a discrete contingency space $\mathscr{X}$, a pre-image function $PreImage : \mathbb{R}^n \to \mathscr{X}$, an objective function $O : \mathscr{X} \to \mathbb{R}$ and a threshold $\gamma \in \mathbb{R}$.

Algorithm parameters: the parameters $\lambda_0 = [\mu_0, \Sigma_0]$ of the initial n-dimensional Gaussian sampling distribution, the size s of the sample drawn at each iteration and the number m of best solutions chosen at each iteration.
Input: the amount $res_{available}$ of available computational resources.
Output: a set $\mathscr{X}_{dang}$ of elements of $\mathscr{X}$ such that $O(x) \geq \gamma$.
Algorithm:

Step 1. Set $t = 1$ and $res = res_{available}$.
Step 2. Set P and $\mathscr{X}_{dang}$ to empty sets.
Step 3. While $res > 0$, **do:**
$(\mathscr{X}_{dang}, P, res) = BIS(\mathscr{X}_{dang}, P, res)$ (see Figure 3.7)
$t \leftarrow t + 1$
end while
Step 4. Output $\mathscr{X}_{dang}$ and stop.

Figure 3.8: Fully specified version of a comprehensive iterative sampling algorithm for identifying the elements such that a function $O : \mathscr{X} \to \mathbb{R}$ exceeds a threshold γ when the available computational resources are bounded, $\mathbb{R}^n$ being chosen as Euclidean embedding space.

3.5 Discussion

The comprehensive iterative sampling algorithm that has been provided on Figure 3.8 allows to identify dangerous contingencies within a discrete contingency space, provided that there exists a procedure allowing to embed this space in a Euclidean space. Two examples of such a procedure have been proposed in the case where the considered contingency space is exclusively composed of $N-k$ line outage contingencies. This framework can easily be extended to other kinds of contingency spaces by adapting the metrization procedure to the performed study.

For instance, if the contingency space gathers $N-k$ contingencies combining different types of equipment outages, it can also be embedded in $\mathbb{R}^{2k}$ by projecting each single equipment outage contingency in the plane according to the equipment's coordinates (either geographical or electrical), and by using the same pre-image function as the one defined in this chapter for $N-k$ line outage contingencies.

When considering other types of contingencies, such as shifts in the load pattern, a measure of distance has to be defined between each pair of contingencies, for example by computing the Euclidean distance between vectors concatenating post-contingency system state variables. The multidimensional scaling algorithm provided in this chapter can then be used for computing coordinates for the contingencies in a low-dimensional Euclidean space, on which the comprehensive iterative sampling algorithm can be executed. In this case, the pre-image function would simply associate to a point of this latter space the nearest contingency.

4

Case studies

In this chapter, the iterative sampling approach that was proposed in Chapter 3 is illustrated on three different case studies. We first report the results of an $N-3$ analysis performed on the IEEE 118 bus test system, followed by those of $N-1$ and $N-2$ analyses carried out on the Belgian transmission system.
The simulation results presented in this Chapter have been published in [37] and [38].

4.1 Results on the IEEE 118 bus test system for $N-3$ security analysis

4.1.1 Problem

In this section, we apply the proposed methodology on the IEEE 118 bus test system (described in [39]), which has been vastly used as benchmark test system in the literature.

The contingency space considered in this study is the set of all potential $N-3$ "line outage" contingencies. The IEEE 118 bus test system counting 186 branches, there exist $\binom{186}{3} = 1\,055\,240$ such contingencies.

We define the objective function as the number of iterations needed by an AC load-flow (using Newton's method) to converge when applied to the post-contingency situation. The motivations for such a choice have been described earlier in Chapter 2.5.2.4. If, for some contingencies, the load-flow algorithm diverges or does not converge after the maximal number of iterations (10) has been reached, the corresponding value of the objective function is set equal to the maximal number of iterations plus one (11).

The threshold γ on the values of the objective function above which a contingency is considered as dangerous is set equal to 11. In other words, a contingency is defined as dangerous if an AC load-flow run on the post-contingency situation diverges or has not converged after 10 iterations. In order to evaluate the results of our approach, we first screened the whole contingency space and thus found out that there were 187 dangerous contingencies. The ratio between the dangerous contingencies and all possible contingencies is equal to $1.77 \cdot 10^{-4}$.

The pre-image function we adopt for embedding the contingency space in a Euclidean space is the one that was defined in Chapter 3.3 (based on the inter-bus electrical distances). The space of the considered $N-3$ line outage contingencies is thus embedded in $\mathbb{R}^6$.

The simulation results presented hereafter have been obtained with the following algorithm parameters: the number s of points drawn from the contingency space at each iteration is set equal to 200 and the number m of points with highest values of the objective function that are used to define the sampling distribution to be used during the next iteration is chosen equal to 10. These values follow the recommendations provided in Chapter 2.3, according to which the parameter s should be one order of magnitude larger than the number of elements parametrizing the sampling distributions the algorithm works with (equal to 42 here since the mean vectors parametrizing Gaussian distributions defined on $\mathbb{R}^6$ comprise 6 elements and the covariance matrices 36 ele-

ments) and the parameter m should be 10 to 20 times smaller than s. We chose among the range of values satisfying these conditions those yielding the best performances.

The maximal number i_{max} of iterations to be done during each "sub-run" of the iterative sampling algorithm if none of the other stopping conditions is reached is set equal to 60.

4.1.2 Simulation results

We will mostly use this case study to illustrate how rapidly our approach navigates in the contingency space towards a dangerous contingency with respect to a naive Monte Carlo sampling procedure.

To show the efficiency of our methodology, we have studied the speed at which it can identify one single dangerous contingency when assuming that the available computational resources are unlimited. This speed has then been compared with the one corresponding to a classical Monte Carlo sampling of the event space. For the iterative sampling method, we ran the algorithm 100 times and stored for each of them the number of contingencies that had been screened when the first dangerous one was found. For the Monte Carlo sampling method, we took 100 random permutations of the $1\,055\,240\ N-3$ contingencies and studied for each of these permutations the number of contingencies to screen before encountering the first dangerous one. The results of these simulations are collected in the histograms reported in Figure 4.1 for our iterative sampling algorithm and Figure 4.2 for the Monte Carlo sampling method. The height of each vertical bar on these figures corresponds to the number of runs of the considered sampling method during which the first dangerous contingency was identified after having screened a number of contingencies belonging to the range specified on the horizontal axis.

We observe that the number of contingencies screened before identifying the first dangerous one is centered around 1 403 with a standard deviation equal to 941 for the iterative sampling algorithm, and centered around 5 070 with a standard deviation equal to 4 553 for the Monte Carlo method. The average number of contingencies screened when the first dangerous one is found is more than 4 times smaller for our iterative sampling method, which means that it is significantly more efficient than the classic Monte Carlo method as regards the search of one dangerous contingency. This result is explained by the fact that our approach exploits during each iteration $i > 1$ the information contained in the previously drawn sample to compute a new sampling distribution which is more likely to give more weight to events leading to high values of O.

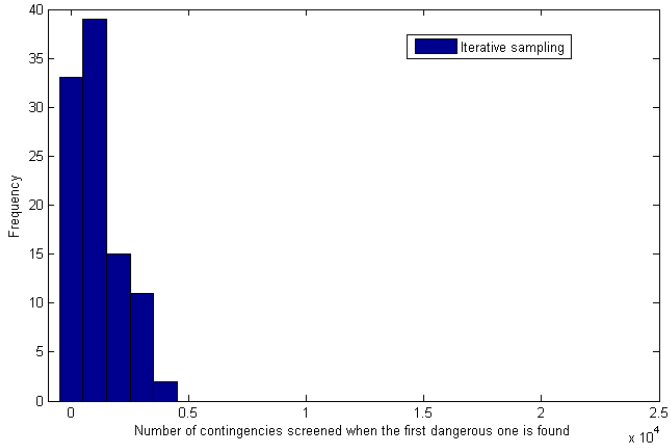


Figure 4.1: Repartition of the number of contingencies screened before identifying the first dangerous one over 100 runs of our iterative sampling algorithm.

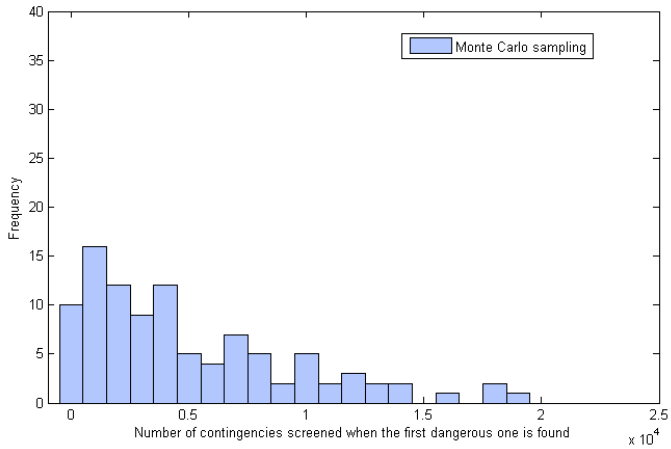


Figure 4.2: Repartition of the number of contingencies screened before identifying the first dangerous one over 100 runs of the Monte Carlo sampling method.

This property of our algorithm is illustrated on Figure 4.3, which shows the average values taken by the objective function on the successive samples drawn during a typical execution of the first "sub-run" of our iterative sampling approach. The horizontal axis of this figure represents the number of the iteration while each vertical bar represents the average plus or minus the standard deviation of the values of the objective function in the sample of 200 points generated at a given iteration.

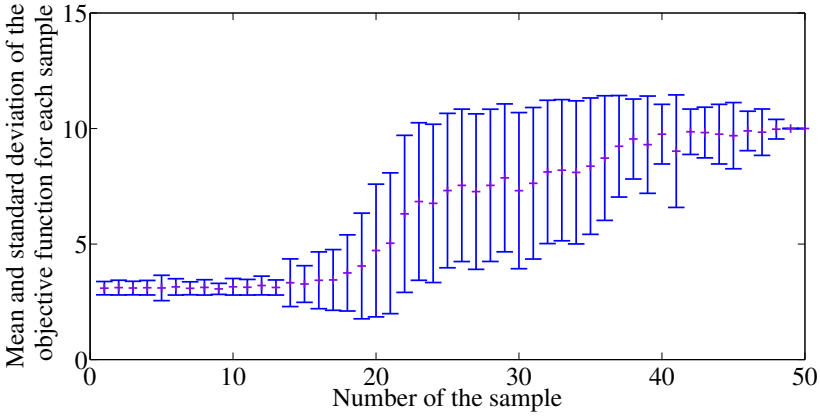


Figure 4.3: Evolution of the range of values that the objective function takes on the successive 200 point samples generated during a typical execution of the first "sub-run" of our comprehensive iterative sampling approach.

4.2 Results on the Belgian transmission system: $N-1$ analysis

4.2.1 Problem

This section presents the performances of the proposed approach on real data. We consider here the Belgian transmission system, and more specifically its equipments operated at a voltage level of 150 kV and above (i.e., 597 buses and 635 transmission lines).

The contingency space in which we want to identify dangerous contingencies includes all the $N-1$ line tripping contingencies, the initial operating conditions being the same for all these contingencies.

The objective function adopted in this analysis relies on an equipment-based criterion. As proposed in Chapter 2.5.3.2, we choose to focus on one single equipment, the line Bruegel-Courcelles (2) 380 kV (a non-border 380 kV transmission line between Bruegel and Courcelles). The value of the objective function for a given contingency is defined as being equal to the loading rate (expressed as the percentage of the maximal admissible loading rate) of this target transmission line in the post-contingency steady-state. The security analysis tool we adopt to simulate these post-contingency steady-states is a DC load-flow, whose complexity is well adapted to the problem treated here.

The threshold γ on the value of the objective function defining the dangerous contingencies is set equal to 22.4 (it is a percentage), which is twice the value of the loading rate of line Bruegel-Courcelles (2) 380 kV in the base case, equal to 11.2 %.

Given our definitions of O and γ, the contingency space comprises 634 contingencies and there are 6 dangerous contingencies among them (this latter number can be computed by simulating each of the potential contingencies). These dangerous contingencies are presented on Table 4.1, in which they are sorted by decreasing severity and numbered from 1 to 6 for easier referral. Note that the ratio between the number of dangerous contingencies and the total number of contingencies is equal to $9.46 \cdot 10^{-3}$.

The contingency space is embedded in the plane by following the procedure explained in Chapter 3.2 for embedding a set of $N-1$ line outage contingencies in $\mathbb{R}^2$. Each contingency is thus projected on the plane as the midpoint of the segment representing the disconnected transmission line in the geographical map of the Belgian transmission system. The pre-image function associates to each point of the plane the closest projected contingency.

Figure 4.4 presents the profile of the objective function on this embedding space. The color scale goes from dark blue for its lowest values to dark red for the highest ones. Each colored surface corresponds to the "influence zone" (which is actually a

Table 4.1: List of the dangerous contingencies with respect to the transmission line Bruegel-Courcelles (2) 380 kV, sorted by decreasing severity, where the severity of a contingency is the loading rate it induces on the target transmission line.

Dangerous contingency number	Severity (loading rate of line Bruegel-Courcelles (2) 380 kV)
1	32.2 %
2	26.1 %
3	25.9 %
4	25.4 %
5	25.1 %
6	22.6 %

Voronoi cell) of a contingency, i.e. the area in which each point is associated to the same contingency by the pre-image function. This profile has been built by evaluating the objective function for all the $N-1$ contingencies. The areas of the map corresponding to the 6 dangerous contingencies are those colored in shades of orange and red. Note that the profile of the objective function is reported here for information but is not an input of the problem. The targeted transmission line is also represented on this figure.

As a complement of information, Figure 4.5 shows the projection of all the considered $N-1$ contingencies on the plane as the midpoint of the tripped line. The red crosses correspond to the dangerous contingencies, numbered as in Table 4.1, and the black diamond locates the midpoint of the target transmission line.

The algorithm parameters adopted to generate the simulations reported in the next section are the following: the number s of points drawn from the contingency space at each iteration is set equal to 50 and the number m of points with highest values of the objective function that are used to define the sampling distribution to be used during the next iteration is set equal to 5.

Here again, these values follow the recommendations provided in Chapter 2.3: the number of elements parametrizing the sampling distributions computed by the algorithm being equal to 6 in this problem (since a Gaussian distribution defined on $\mathbb{R}^2$ is parametrized by a 2-by-1 mean vector and a 2-by-2 covariance matrix), we chose s one order of magnitude larger and m 10 times smaller than s. The maximal number i_{max} of iterations to be done during each "sub-run" of the iterative sampling algorithm if none of the other stopping conditions is reached is set equal to 10.

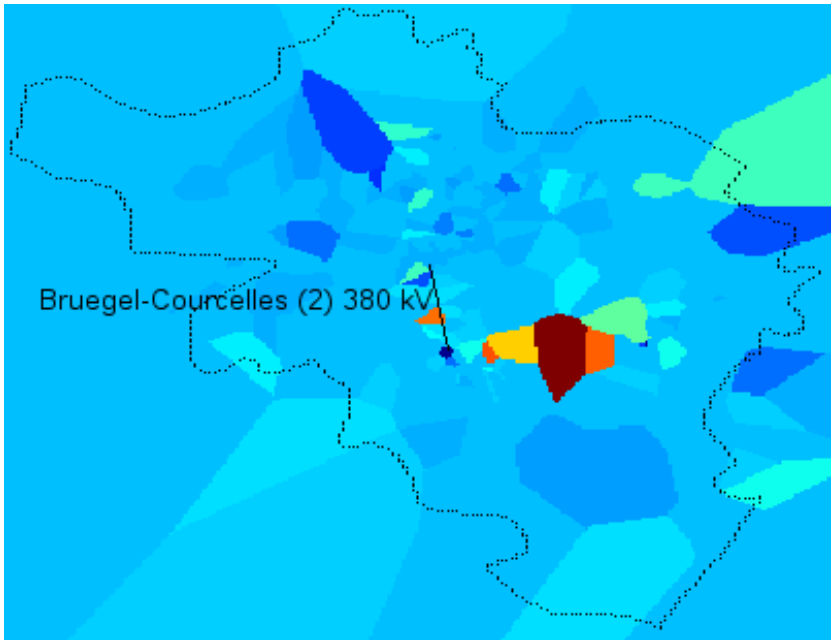


Figure 4.4: Profile of the objective function over the Euclidean embedding space.

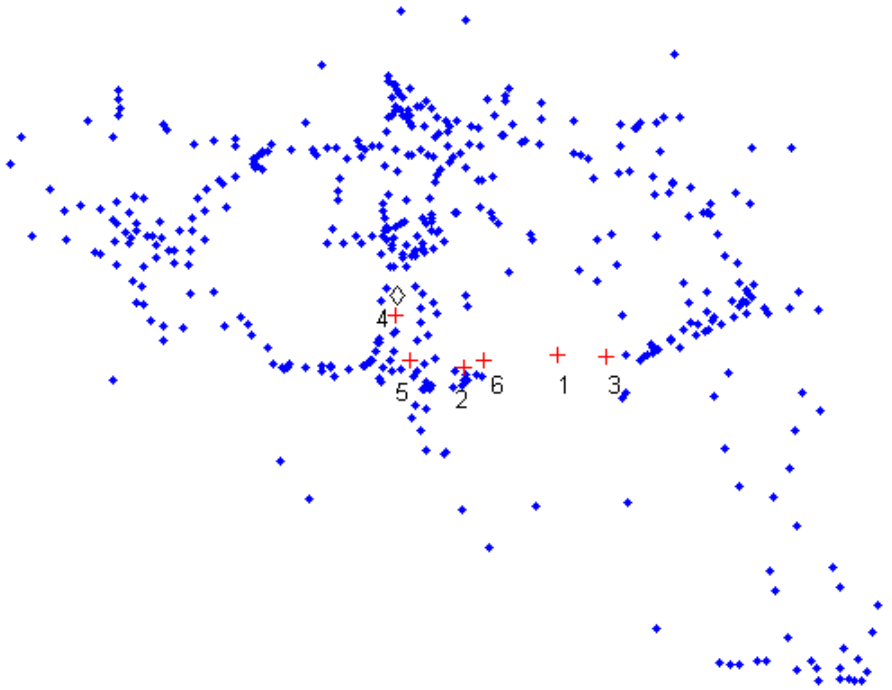


Figure 4.5: Representation of all $N-1$ contingencies (by the midpoint of the lost line) in $\mathbb{R}^2$. The dangerous contingencies are represented by the red crosses. The black diamond represents the midpoint of line Bruegel-Courcelles (2) 380 kV, chosen as target transmission line.

4.2.2 Implementation details

The model of the Belgian transmission system used in for this study was provided by Elia, the Belgian TSO. The DC load-flow tool we worked with is also the one used by Elia. It is one of the functionalities of their security assessment software, Plaire, that was developed by Tractebel.

We used a script written in Perl in order to run this tool in batch mode and thus compute the value of the objective function for any given contingency. Our iterative

sampling algorithm has been implemented in Python and internally calls this Perl script when it is necessary to evaluate the objective function for a contingency.

4.2.3 Simulation results

We will first study in this subsection a typical run of the developed comprehensive iterative sampling algorithm applied to the problem described previously, with a computational budget equal to the total number of contingencies in the contingency space (634) so as to have a clear view of the behavior of the algorithm. The results of the first "sub-run" of this typical run are presented on Figure 4.6. The black points on each sub-figure represent the set of 50 points drawn from the search space during, respectively, the first 6 iterations. Among these points, those corresponding to dangerous contingencies are represented as red crosses.

These figures show that the successive sampling distributions built at each iteration of the first "sub-run" of the algorithm rapidly concentrate to an area corresponding to the dangerous contingency number 1. Moreover, the dangerous contingencies number 3 and 6, located in the neighborhood of this latter one, have also been identified.

The following "sub-run" of this typical run has converged towards one of the other dangerous contingencies not yet identified, and did not come across any other dangerous contingency during its execution. The third "sub-run" only saw dangerous contingencies that had already been seen and no new one. The last two dangerous contingencies have been identified during the fourth sub-run, after having screened 403 contingencies since the launching of the algorithm. These results show that re-running the basic iterative sampling algorithm several times allows to leverage the performances of one single run of this basic algorithm, and to exploit the fact that this basic algorithm can converge towards different maxima of the objective function from one execution to the other.

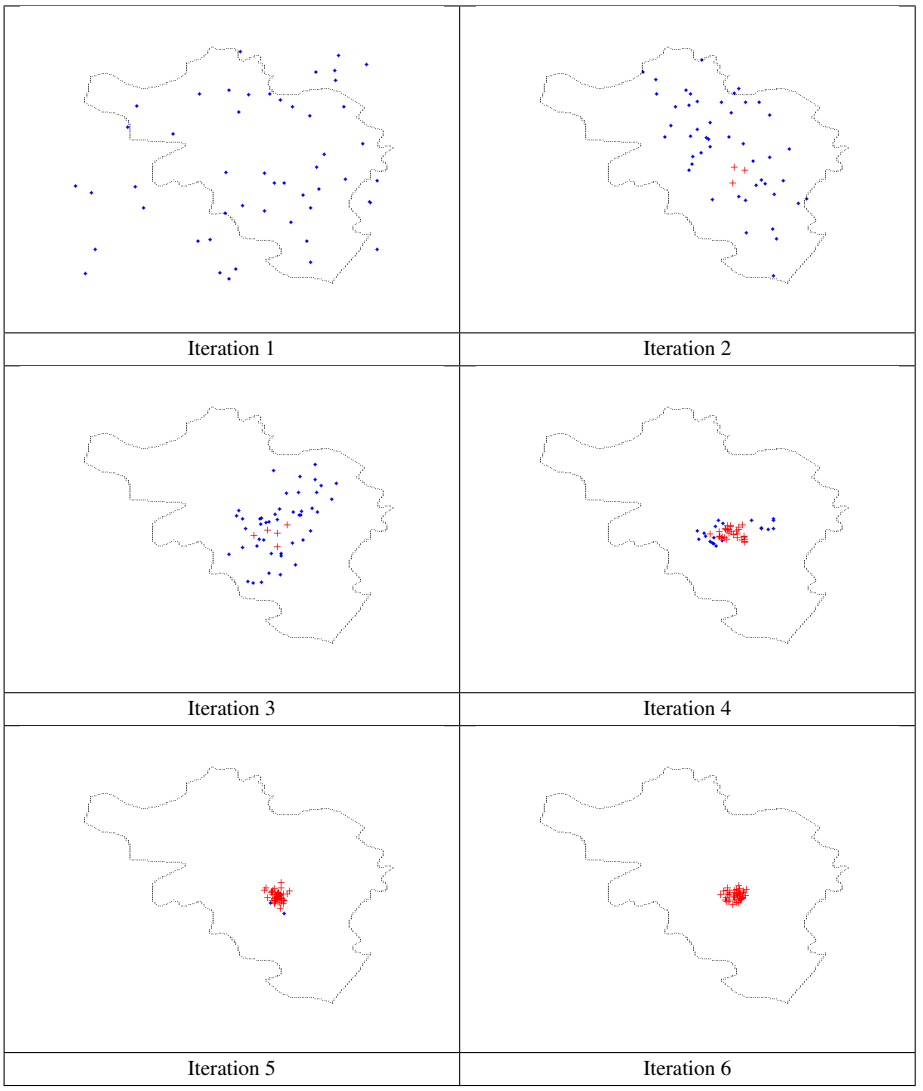


Figure 4.6: Representation of the successive samples of 50 points drawn from the contingency space over the iterations during the first sub-run of the comprehensive iterative sampling algorithm.

We now propose to study the number of dangerous contingencies our approach is able to identify depending on the allocated computational budget. To do so, we ran our iterative sampling framework as well as a classical Monte Carlo sampling approach 100 times with various computational budgets comprised between 0 and 634 possible evaluations of the objective function, and plotted on Figure 4.7 the mean of the number of dangerous contingencies identified over these 100 runs versus the amount of available computational resources. Figures 4.8 and 4.9 also show the standard deviation of the number of dangerous contingencies identified over 100 runs of our iterative sampling approach and of a classical Monte Carlo sampling algorithm, respectively.

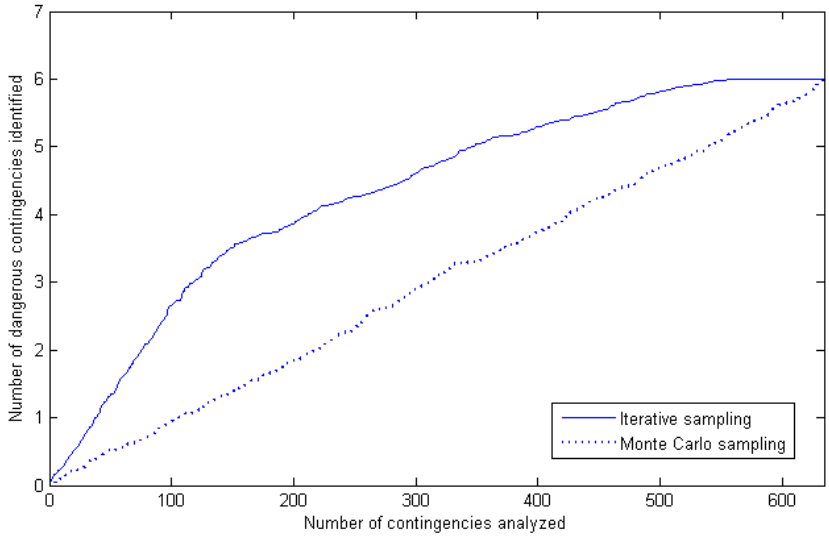


Figure 4.7: Mean of the number of dangerous contingencies identified over 100 runs of our iterative sampling approach (solid line) and of a classical Monte Carlo sampling (dotted line) versus the available computational budget, when considering an $N-1$ security analysis where the objective function is the loading rate of line Bruegel-Courcelles (2) 380 kV.

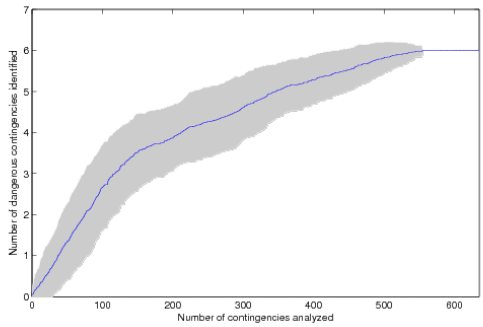


Figure 4.8: Mean (solid line) plus and minus standard deviation (grey area) of the number of dangerous contingencies identified over 100 runs of our iterative sampling approach versus the available computational budget, when considering an $N-1$ security analysis where the objective function is the loading rate of line Bruegel-Courcelles (2) 380 kV.

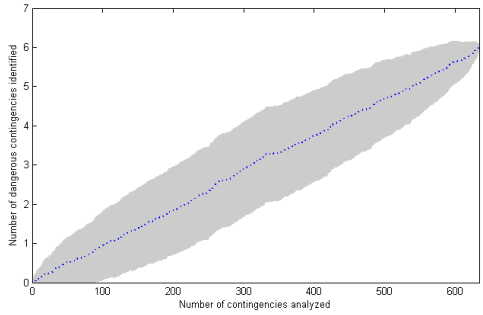


Figure 4.9: Mean (dotted line) plus and minus standard deviation (grey area) of the number of dangerous contingencies identified over 100 runs of a classical Monte Carlo sampling versus the available computational budget, when considering an $N-1$ security analysis where the objective function is the loading rate of line Bruegel-Courcelles (2) 380 kV.

This figures shows that our iterative sampling approach identifies on average a much higher number of dangerous contingencies than a classical Monte Carlo sampling (for instance, twice more when the amount of available computational resources is set to 200 contingency analyses), especially when the available computational budget is lower than half the size of the contingency space. As explained before, this is due to the fact that, during each of its "sub-runs", our approach uses the information contained in the data sampled from the contingency space so as to direct the search towards the dangerous contingencies over the iterations. Moreover, the standard deviation of the number of dangerous contingencies identified is slightly smaller for our iterative sampling method than for a classical Monte Carlo sampling method.

In the following, we will adopt a computational budget of 150 contingency analyses, which is a relevant setting to reproduce the context where the available computational resources are bounded and do not allow to screen exhaustively the whole contingency space.

Table 4.2 presents the probabilities of identification of the six dangerous contingencies in this context. These probabilities have been computed over 100 runs of the iterative sampling algorithm by counting in how many runs each of these contingencies were identified. The performances of the proposed procedure are compared in this table to those of a classical Monte Carlo sampling of the contingency space. For this latter method, we took 100 random sets of 150 $N-1$ contingencies from the 634 potential ones, and counted how many times each of the six dangerous contingencies appeared in these sets. The results collected in Table 4.2 correspond to the conversion of these numbers into probabilities.

Table 4.2: Probabilities of identification of the six dangerous contingencies (w.r.t. the transmission line Bruegel-Courcelles (2) 380 kV) during one run.

Contingency number	**Probability of identification**	
	Iterative sampling	Monte Carlo
1	0.91	0.22
2	0.41	0.14
3	0.74	0.21
4	0.36	0.22
5	0.51	0.29
6	0.83	0.26

We observe that our iterative sampling framework allows to identify the most dangerous $N-1$ contingency with a very satisfying probability (0.91), two others of the six dangerous contingencies with a probability greater than 0.7, and the three last ones with a probability of 0.36 to 0.51. Even these last figures are still higher than the performances of the Monte Carlo method, with which all the contingencies are identified with a probability 0.20 in expectation.

In addition to these results, Table 4.3 shows the probabilities of identifying at least n dangerous contingencies (when n varies between 1 and 6) with the iterative sampling algorithm and the Monte Carlo method. These probabilities are still computed over 100 runs for each method and with a computational budget of 150 contingency analyses.

Table 4.3: Probabilities of identifying at least n dangerous contingencies (w.r.t. transmission line Bruegel-Courcelles (2) 380 kV) during one run of our iterative sampling approach.

n	**Probability of identifying at least n dangerous contingencies**	
	Iterative sampling	Monte Carlo
1	0.99	0.69
2	0.90	0.31
3	0.76	0.09
4	0.47	0.04
5	0.20	0
6	0.04	0

The probabilities of identifying at least n dangerous contingencies are much higher with our approach than with a Monte Carlo sampling (1.4 to 10 times better for $n = 1$ to 4). Note that none of the 100 runs of a Monte Carlo sampling method that were performed to compute these statistics allowed to identify 5 or 6 dangerous contingencies.

Finally, we can add to these results that the average number of dangerous contingencies identified during one run of the iterative sampling method with a computational budget of 150 contingency analyses is equal to 3.4, which shows a large improvement with respect to the 1.1 dangerous contingencies identified on average by the Monte Carlo method with the same amount of computational resources.

All these results highlight the interest of our importance sampling approach, which is able to identify the contingencies that are dangerous for a target transmission line

with a rather high probability and while screening less than one fourth of the whole contingency space.

4.3 Results on the Belgian transmission system: $N-2$ analysis

4.3.1 Problem

We now focus on the problem of an $N-2$ security analysis, also on the part of the Belgian transmission system operated at 150 kV and above. This network includes 635 transmission lines, and there are therefore $\binom{635}{2} = 201\,295$ potential $N-2$ contingencies.

The objective function adopted in this study is a global criterion inspired from the common engineering practice of the Belgian TSO. For a given contingency, it is set equal to the maximal loading rate (expressed as a percentage) observed over all the transmission lines of the system in the post-contingency steady-state. To evaluate its value, the post-contingency steady-state is first simulated using a DC load-flow, so as to determine the loading rate induced by the considered contingency on all the lines that remain connected. The maximal value among these loading rates is then computed.

A contingency is considered as dangerous if its value of the objective function exceeds $\gamma = 170$, i.e. if it induces on any line of the network an overload greater than 170 %. An exhaustive analysis of the contingency space has shown that there are 210 such dangerous contingencies. The ratio between the number of dangerous contingencies and the total number of contingencies is equal to $1.04 \cdot 10^{-3}$.

Here again, mainly for interpretability reasons, the contingency space is embedded in a Euclidean space by following the procedure proposed in Chapter 3.2, based on the geographical coordinates of the system equipments. As we consider $N-2$ (line outage) contingencies here, $\mathbb{R}^4$ is used as Euclidean embedding space.

The algorithm parameters chosen for this study are the following: the size s of each sample drawn by the current sampling distribution is set equal to 100 (one order of magnitude larger than the number of elements parametrizing Gaussian distributions defined on $\mathbb{R}^4$, equal to 20), and the number m of best scoring contingencies used to define at the end of each iteration the parameters of the next sampling distribution is set equal to 10 (10 times smaller than s).

The maximal number i_{max} of iterations to be done during each "sub-run" of the iterative sampling algorithm if none of the other stopping conditions is reached is set equal to 20.

The other implementation details are the same as those of the $N-1$ analysis presented in the previous section.

4.3.2 Simulation results

To study the performances of our iterative sampling approach on this setting, we first compared the empirical mean and standard deviation (over 100 runs of our iterative sampling algorithm) of the number of dangerous contingencies it identifies given different computational budgets. These results are collected in Figure 4.10, in which they are compared with the ones obtained by a classical Monte Carlo sampling approach. For this latter method, we drew 100 random sets of a number of different contingencies corresponding to the considered computational budget, counted how many dangerous contingencies appeared in each of these sets and plotted the mean and standard deviation of these numbers.

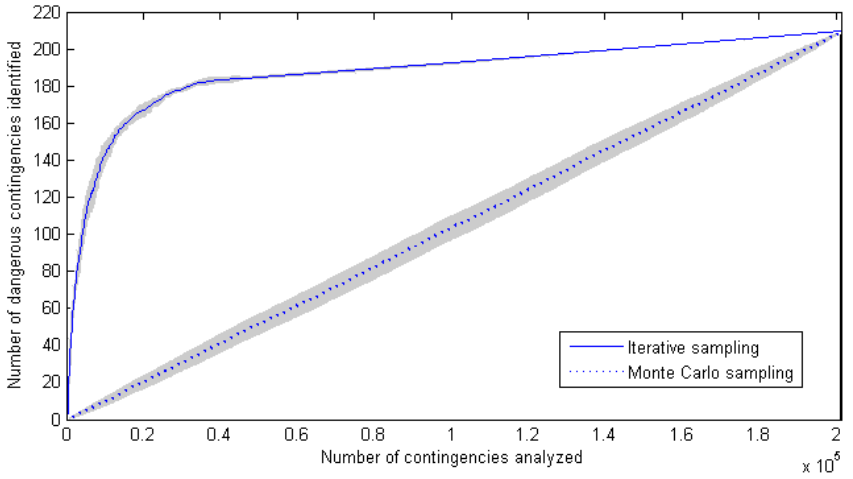


Figure 4.10: Mean (solid line for our iterative sampling approach, dotted line for a classical Monte Carlo sampling) plus and minus standard deviation (grey areas) of the number of dangerous contingencies identified over 100 runs of each method versus the available computational budget, when considering an $N-2$ security analysis where the objective function is the maximal overload induced by a contingency on all the lines of the sytem.

We observe that the our iterative sampling approach brings a significant improvement to the results of a classical Monte Carlo sampling. Indeed, it allows to identify up to 60 times more dangerous contingencies, with a much lower standard deviation. The iterative sampling algorithm especially outperforms the Monte Carlo sampling when the available computational budget is low with respect to the size of the contingency space (up to 25 smaller).

To deepen the analysis of the performances of our iterative sampling framework, we also computed an empirical estimate of probability of identifying at least n dangerous contingencies with both this framework and a Monte Carlo sampling method. These results are reported in Table 4.4.

Table 4.4: Probabilities of identifying at least n dangerous contingencies while analyzing 750 different contingencies with our iterative sampling approach and by screening the same number of contingencies picked up from the contingency space using the Monte Carlo method (these probabilities are computed empirically from 100 runs of each approach).

n	**Probability of identifying at least n dangerous contingencies**	
	Iterative sampling	Monte Carlo
1	1	0.49
2	1	0.20
3	1	0.03
4	1	0.01
5	0.99	0
10	0.95	0
20	0.75	0
30	0.51	0
40	0.26	0
50	0.13	0
100	0	0
210	0	0

These results show the ability of our iterative sampling approach to identify at least 30 dangerous contingencies with a high probability (and up to more than 50 of them with lower probabilities), while a classical Monte Carlo sampling method is unable to identify more than 4 of them while screening the same number of contingencies.

5

On-line selection of iterative sampling algorithms

In the comprehensive iterative sampling algorithm developed in this thesis, we have chosen to repeat the same basic iterative sampling algorithm as long as there were some computational resources available in order to identify a maximal number of dangerous contingencies. We now assume that several basic iterative sampling algorithms are available. The differences between such algorithms can for instance come from the choice of some parameters like the initial sampling distribution, the number of points in a sample or the number of best solutions used at the end of each iteration to compute the next sampling distribution. Considering that these algorithms are executed sequentially, this chapter proposes an on-line method for determining which one to call so as to maximize the number of different dangerous contingencies identified over the sequence of calls (and while respecting the available computational budget).
We will show that the method we propose for reaching this goal shares some similarities with algorithms that are used to solve multi-armed bandit problems.

5.1 Introduction

The choice of the parameters of a basic iterative sampling algorithm, and in particular the choice of the initial sampling distribution, strongly influences the obtained results. For instance, if the initial sampling distribution of a basic iterative sampling algorithm focuses on a small subpart of the plane, it is more likely that this algorithm identifies only some dangerous contingencies localized in this specific area. To a lesser extent, the values of the parameters s and m also affect the performances of the algorithm. All these parameters are usually chosen in a heuristic fashion and/or based on expert knowledge so that the algorithm yields good performance.

We choose in this chapter to work not only with one single basic iterative sampling algorithm, but rather with several of them. Each such algorithm uses different sets of parameters seeming relevant with respect to the security assessment problem at hand. In particular, an interesting way to define a set of basic iterative sampling algorithms on a given problem consists in splitting the contingency space into several sub-areas and defining as many algorithms as there are such sub-areas, with initial sampling distributions respectively focused on each of these sub-areas. All these algorithms share the same parameters s and m, and of course, are applied to the same problem (i.e., with the same contingency space, pre-image function, objective function and threshold γ). From a technical point of view, they are implemented as explained previously in Figure 3.7.

In this context where several different iterative sampling algorithms are available, we focus on the problem of selecting in a dynamic way iterative sampling algorithms from this set so as to maximize the number of different dangerous contingencies identified over a finite sequence of runs of these methods.

One simple illustration to show the interest of selecting iterative sampling algorithms in a dynamic way could be the following. If, after several runs of one algorithm of the pool, we observe that always the same dangerous contingencies are identified, it would then be wise to stop exploiting this algorithm and to switch to another one.

Note that, in the case where the available iterative sampling algorithms only differ by their initial sampling distributions, the selection strategy proposed in this chapter is another way to deal with the potential multimodality of the objective function. Running different iterative sampling algorithms, each of them focusing on one specific subpart of the search space, indeed maximizes the probability to come across all the local maxima of the objective function.

5.2 Problem formulation and sketch of our solutions

We consider here the case where we have a set of n iterative sampling algorithms IS^1, IS^2, ..., IS^n, and that the available computational resources are bounded. The problem we want to address is the scheduling of the calls to these algorithms so as to identify, before the computational resources are exhausted, as many different dangerous contingencies as possible.

We will assume here that only one iterative sampling algorithm can be run at a time. The next one can only be launched when the previous one has reached its terminal conditions and outputted its results. Therefore, a scheduling strategy can possibly exploit information obtained from the previous runs so as to decide which new algorithm to call.

In the following, we will explore two (families of) strategies for scheduling these calls:

A strategy looping over the iterative sampling algorithms. Such a strategy calls sequentially the n algorithms IS^1, IS^2, ..., IS^n. Once this whole sequence has been run, it starts again to call the same sequence, and so on and so forth until the computational resources are exhausted.

A discovery rate-based strategy. This strategy will score each iterative sampling algorithm according to its ability to discover new dangerous contingencies, and sequentially pick up the strategy with the highest score as long as some computational resources are available.

5.3 Detailed algorithm: strategy looping over the available set of iterative sampling algorithms

Figure 5.1 provides a fully specified algorithm to schedule a sequence of calls of the available iterative sampling algorithms by looping over them and while respecting the computational resources available. This strategy calls successively the n available iterative sampling algorithms, stores the dangerous contingencies identified and stops when the available computational resources have been exhausted.

Problem definition: a contingency space $\mathscr{X}$, a pre-image function *PreImage* : $\mathbb{R}^n \to \mathscr{X}$, an objective function $O : \mathscr{X} \to \mathbb{R}$ and a threshold $\gamma \in \mathbb{R}$.
In addition to these usual characteristics of the dangerous contingency search problem addressed in this thesis, we assume that a set of n iterative sampling algorithms IS^1, IS^2, ..., IS^n defined on the problem stated before is available. Each of these algorithms is characterized by its own parameters λ_0, s and m (chosen by the user).

Input: the amount $res_{available}$ of available computational resources (i.e., the maximal number of times the objective function can be evaluated).
Output: a set $\mathscr{X}_{dang}$ of elements of $\mathscr{X}$ such that $O(x) \geq \gamma$.
Algorithm:

Step 1. Set $i = 1$ and $res = res_{available}$.
Step 2. Set $\mathscr{X}_{dang}$ and P to empty sets.

Step 3. While $res > 0$ **do:**
$(\mathscr{X}_{dang}, P, res) = IS^i(\mathscr{X}_{dang}, P, res)$ (see Figure 3.7).
If $i < n$ **then:** $i \leftarrow i + 1$
else: $i = 1$
end if
end while

Step 4. Output $\mathscr{X}_{dang}$ and stop.

Figure 5.1: A detailed strategy for looping over a set of iterative sampling algorithms while respecting a given budget of computational resources.

5.4 Detailed algorithm: discovery rate-based strategy

Whereas the previous strategy scheduled the calls of the available iterative sampling algorithms by repeating them sequentially in the same order, we now propose a strategy that analyzes the results obtained after each algorithm call in order to schedule these calls in a more efficient way.

This strategy works as follows. It first calls once each algorithm of the available set to perform initialization. Then, at each step $t > n$, it computes for each algorithm IS^i an index $D^i_{t-1} \in \mathbb{R}$ evaluating its performances over the previous runs. If $d^i_T(t)$ is the number of *new* dangerous contingencies identified by algorithm IS^i during its last T executions, computed at step t, and $n^i(t)$ the number of times algorithm IS^i has been called up to step t included, this index (that we name "discovery factor") is defined as

follows:

$$D^i_{t-1} = \begin{cases} d^i_T(t-1) \text{ if } T \leq n^i_{t-1} \\ d^i_{n^i(t-1)}(t-1) \text{ if } T > n^i_{t-1} \end{cases} \tag{5.1}$$

The discovery factor of algorithm IS^i represents the number of new dangerous contingencies it has been able to identify over its T previous runs. It has been designed based on the following empirical observation we made: *given a sequence of* $t-1$ *runs of an iterative sampling algorithm, the run* t *is likely to identify new dangerous contingencies if the most recent runs (say, the last* T *runs) have been able to identify new dangerous contingencies.*

In the simulations we have carried out, we have observed that the value of $T = 2$ was giving the best results. Very large values of T did not perform well at all probably because they were biasing too much the choice of the IS algorithm to run towards the one that had identified the largest number of dangerous contingencies over the whole process, even if it was unable anymore to identify new ones.

Like the procedure presented in the previous section, this strategy stores the dangerous contingencies and stops when the available computational resources are exhausted. The fully specified algorithm corresponding to this discovery rate-based strategy is presented on Figure 5.2.

Problem definition: a contingency space $\mathscr{X}$, a pre-image function $PreImage$: $\mathbb{R}^n \to \mathscr{X}$, an objective function $O : \mathscr{X} \to \mathbb{R}$ and a threshold $\gamma \in \mathbb{R}$.
In addition to these usual characteristics of the dangerous contingency search problem addressed in this thesis, we assume that a set of n iterative sampling algorithms IS^1, IS^2, ..., IS^n defined on the problem stated before is available. Each of these algorithms is characterized by its own parameters λ_0, s and m (chosen by the user).

Input: the amount $res_{available}$ of available computational resources (i.e., the maximal number of times the objective function can be evaluated).
Output: a set $\mathscr{X}_{dang}$ of elements of $\mathscr{X}$ such that $O(x) \geq \gamma$.
Algorithm:

Step 1. Set $t = 1$ and $res = res_{available}$.
Step 2. Set $\mathscr{X}_{dang}$ and P to empty sets.
Step 3. **While** $t \leq n$ and $res > 0$ **do:**

$(\mathscr{X}_{dang}, P, res) = IS^t(\mathscr{X}_{dang}, P, res)$ (see Figure 3.7).
$t \leftarrow t + 1$

end while

Step 4. **While** $res > 0$ **do:**

$t \leftarrow t + 1$
Compute D^i_{t-1} for $i \in \{1, \ldots, n\}$ according to Equation (5.1).
Call IS^k with k such that $k = \arg\max_{1 \leq i \leq n} D^i_{t-1}$. If this maximizer is not unique, draw k at random among the maximizers.

end while

Step 5. Output $\mathscr{X}_{dang}$ and stop.

Figure 5.2: A discovery rate-based algorithm for selecting on-line a sequence of iterative sampling algorithms among the available set so as to maximize the number of different dangerous contingencies identified over this sequence.

5.5 Illustration on the Belgian transmission system

5.5.1 Problem addressed

The simulation results treated in this chapter are related to the power system security assessment problem already addressed in Chapter 4.2: we consider the high voltage Belgian transmission system (equipments operated at 150 kV and above, which represents 600 buses and 635 transmission lines), and we want to identify among all potential $N-1$ line tripping contingencies those that would induce the highest overloads on one specific transmission line.

Here we choose the line named Ruien-Wortegem 150 kV, a 150 kV line between Ruien and Wortegem (whose midpoint is represented by a black diamond on Figure 5.4 in the following section), as target transmission line. We define the value of the objective function for a contingency x as being the loading rate of this target line in the post-contingency steady-state. This loading rate is expressed as a percentage of the maximal flow acceptable on this line.

We consider that a contingency is dangerous if it induces an overload on this line, i.e. a loading rate greater than 100% of the maximal acceptable flow. An exhaustive $N-1$ shows that there are 6 dangerous contingencies (over the 634 potential ones) according to this definition. These dangerous contingencies are listed in Table 5.1 where they are sorted by decreasing severity.

Table 5.1: List of the dangerous contingencies (with respect to transmission line Ruien-Wortegem 150 kV), sorted by decreasing severity (the severity of a contingency is the loading rate it induces on the target transmission line).

Dangerous contingency number	Severity (loading rate of line Ruien-Wortegem 150 kV)
1	124.9 %
2	120.4 %
3	117.1 %
4	115.1 %
5	108.7 %
6	103.8 %

In our simulations, we will purposely work with a large amount of computational resources (we allow 500 contingencies to be analyzed), so that we can examine the be-

havior of the proposed strategies over a long sequence of iterative sampling algorithm executions.

5.5.2 Set of iterative sampling algorithms at hand

In this illustration, we split the plane (modeled hereafter as a rectangle) into 9 sub-areas (numbered as shown on Figure 5.3), and we define 9 different iterative sampling algorithms, respectively initialized in each of these sub-areas.

1	2	3
4	5	6
7	8	9

Figure 5.3: Partition of the contingency space into 9 sub-areas.

Practically, the algorithm IS^k (for $k \in \{1, ..., n\}$) is defined such that its initial sampling distribution is concentrated on sub-area k. The parameters λ_0^i are thus chosen such that μ_0^i corresponds to the geometrical center of sub-area k, and that the jth component of σ_0^i is equal to half the size of sub-area k alongside its jth dimension. Parameters s and m are set respectively to 30 and 3 for all these algorithms.

Figure 5.4 shows, in addition to this partition of the plane, the location of the contingencies in $\mathbb{R}^2$ when representing them by the midpoint of the tripped line. The red crosses correspond to the dangerous contingencies, and the black diamond locates the midpoint of line Ruien-Wortegem 150 kV (chosen as target transmission line).

We observe that 5 of these dangerous contingencies are located in sub-area 1 (only four distinct red crosses are visible since dangerous contingencies 3 and 6 correspond to the loss of lines joining the same nodes) and 1 in sub-area 4. If we knew the position and number of the dangerous contingencies (which is of course not the case in our dangerous contingency search context), we may reasonably expect that running IS^1 and IS^4 would identify them. Here we do not know a priori where the dangerous contingencies are located and, as explained in 5.2, we want to schedule how to successively call these algorithms so as to identify as many dangerous contingencies while respecting the available computational budget.

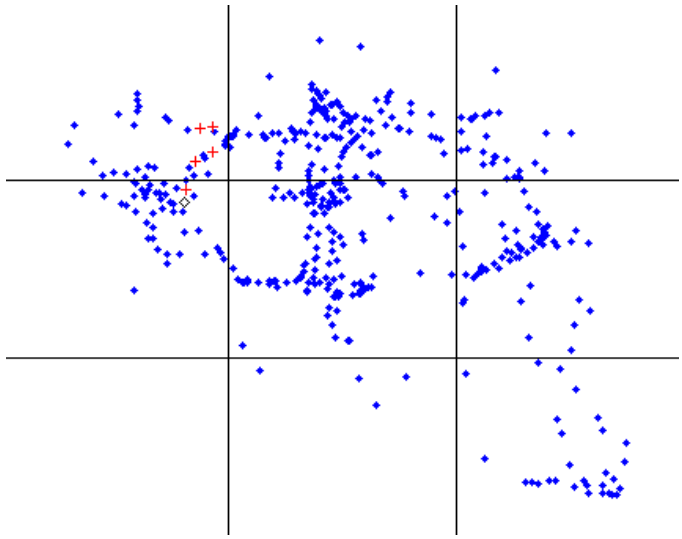


Figure 5.4: Representation of all $N-1$ contingencies (by the midpoint of the lost line) in $\mathbb{R}^2$. The dangerous contingencies are represented by the red crosses. The black diamond represents the midpoint of line Ruien-Wortegem 150 kV, chosen as target transmission line.

5.5.3 Sequential selection strategy looping over the set of iterative sampling algorithms at hand

Table 5.2 presents the results of a particular run of the strategy looping over the set of iterative sampling algorithms available (presented in Section 5.3) on this problem.

We observe that, on the whole, 6 executions of IS^1 and 5 calls of IS^4 are performed before having identified the 6 dangerous contingencies. Note that one execution of these algorithm does not necessarily identify some new dangerous contingencies. As all others iterative sampling algorithms are called in-between, the identification of all the dangerous contingencies requires 46 steps (i.e., 46 calls of an iterative sampling algorithm) of this strategy.

Table 5.2: A particular run of the strategy looping over the available set of iterative sampling algorithms.

Step	Index of the IS algorithm executed	Number of dangerous contingencies identified	Number of contingencies analyzed
1	1	3	26
2	2	3	51
3	3	3	60
4	4	3	115
5	5	3	143
6	6	3	147
7	7	3	115
8	8	3	143
9	9	3	147
10	1	3	151
11	2	3	176
12	3	3	184
13	4	4	199
...	...	...	...
37	1	5	335
...	...	...	...
46	1	6	353

5.5.4 Sequential selection strategy focused on the discovery of new dangerous contingencies

The results of a particular run of the discovery rate-based strategy proposed in Section 5.4 are presented in Table 5.3. Contrary to the strategy looping over the available IS algorithms, this discovery rate-based strategy can start from taking advantage of the previously obtained results from after step 9, the 9 first steps corresponding to the "initialization sequence". As both algorithms IS^1 and IS^4 have identified one new dangerous contingency during their first execution, they have the same discovery factor at step 10 and the algorithm called after this first sequence has to be drawn at random between them. IS^4 is thus executed at step 10, but it does not identify any new dangerous contingency (which could have been expected since there is only one dangerous contingency in sub-area 4). Then IS^1 is selected at step 11 (drawn at random among

IS^1 and IS^4 which still have the same discovery factor) and it finds 3 more new dangerous contingencies. It is executed once again at step 12 since it now has the highest discovery factor, and identifies the 6^{th} dangerous contingency.

With this discovery rate-based strategy for selecting on-line which algorithm to execute, only 12 steps (i.e. 12 executions of an iterative sampling algorithm) have been necessary here before having identified all the dangerous contingencies. On the whole, IS^1 was run 3 times and IS^4 2 times when the 6^{th} dangerous contingency is found. This is better than with the strategy looping over the available algorithms, which had required to analyze 353 contingencies before the 6^{th} dangerous one was found.

Table 5.3: A particular run of the discovery rate-based strategy for selecting on-line which of the available iterative sampling algorithms to execute.

Step	Index of the IS algorithm executed	Number of dangerous contingencies identified	Number of contingencies analyzed
1	1	1	31
2	2	1	55
3	3	1	63
4	4	2	94
5	5	2	136
6	6	2	167
7	7	2	171
8	8	2	176
9	9	2	199
10	4	2	209
11	1	5	220
12	1	6	224

5.5.5 Statistics over 100 runs of these two strategies

Figure 5.5 compares the average performances of the looping and discovery rate-based strategies for on-line iterative sampling algorithm selection. It reports the mean value (computed over 100 runs of each strategy) of the number of dangerous contingencies identified depending on the available computational resources, i.e. the number of contingencies it is possible to analyze.

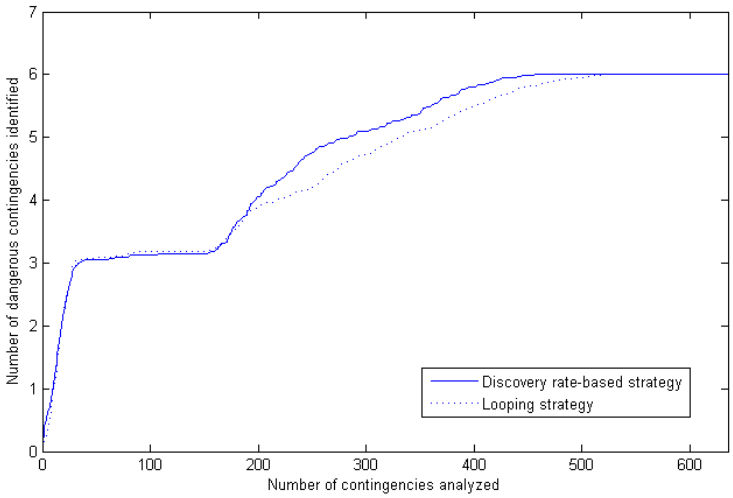


Figure 5.5: Mean value of the number of dangerous contingencies identified over 100 runs of our discovery rate-based (solid line) and looping (dotted line) strategies versus the available computational budget, when considering an $N-1$ security analysis where the objective function is the loading rate of line Ruien-Wortegem 150 kV.

The performances of these two approaches are similar when the computational budget is lower than 150 contingency evaluations since they both call the available iterative sampling algorithms in the same order during their 9 first steps. We observe in particular that the number of dangerous contingencies identified grows very rapidly within the first 35 contingency evaluations, which correspond to the execution of the the algorithm IS^1. This algorithm is indeed the one likely to generate the highest number of dangerous contingencies since four of them are located in sub-area 1. For larger values of the computational budget, the discovery rate-based strategy allows to identify slightly more dangerous contingencies than the looping strategy. This improvement in the results is due to the introduction of the discovery factor to select at each step $t > 9$ the algorithms that obtained the best results in their previous executions.

The standard deviation of the number of dangerous contingencies identified by these two approaches is reported on Figures 5.6 and 5.7. These latter figures show the mean value plus and minus standard deviation (grey area) of the number of dangerous contingencies identified over 100 runs of the discovery rate-based and looping strategies, respectively, versus the available computational budget.

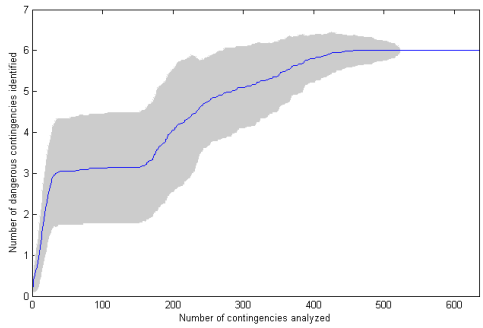


Figure 5.6: Mean (solid line) plus and minus standard deviation (grey area) of the number of dangerous contingencies identified over 100 runs of our discovery rate-based strategy versus the available computational budget, when considering an $N-1$ security analysis where the objective function is the loading rate of line Ruien-Wortegem 150 kV.

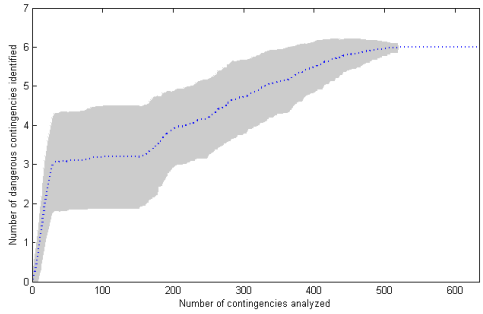


Figure 5.7: Mean (dotted line) plus and minus standard deviation (grey area) of the number of dangerous contingencies identified over 100 runs of our looping strategy versus the available computational budget, when considering an $N-1$ security analysis where the objective function is the loading rate of line Ruien-Wortegem 150 kV.

We observe that, for both strategies, the standard deviation of the number of dangerous contingencies identified is rather high. This is due to the fact that these selection strategies increase the variance of the performances of the basic iterative sampling algorithms that they call.

5.6 Comparison with multi-armed bandit problems

This section highlights the connection that exists between the problem tackled in this chapter and the so-called K-armed bandit problem, that has been well-studied in machine learning and statistics (see [40], [41] and [42]).

5.6.1 Description of the multi-armed bandit problem

A K-armed bandit ($K \in \mathbb{N}$) is a machine learning problem based on an analogy with the traditional slot machine (one-armed bandit) but with more than one arm. Such a problem is defined by the K-tuple $(p_1, p_2, ..., p_K) \in P^K$, P being the set of all reward distributions. When pulled at time $t \in \mathbb{N}$, each arm $k \in \{1, ..., K\}$ provides a reward r_t drawn from a distribution p_k associated with the arm k.

The objective is to maximize the cumulated sum of rewards through iterative pulls. It is generally assumed that no initial knowledge about the arms is available. The crucial trade-off the gambler faces at each trial is between *exploitation* of the arm that has the highest observed reward and *exploration* to get more information about the expected rewards of the other arms. Naturally, the reward distributions $(p_1, p_2, ..., p_K)$ are not supposed to be the same. We denote by $\mu_1, \ldots, \mu_K$ the expected values of the reward distributions $p_1, \ldots, p_K$.

Let $b_t \in \{1, \ldots, K\}$ denote the machine selected at a time t, and let h_t be the history vector available to the gambler at instant t, i.e.:

$$h_t = [b_0, r_0, b_1, r_1, \ldots, b_{t-1}, r_{t-1}].$$

We denote by $\mathscr{H}$ the set of all possible histories of any length.

A policy $\pi : \mathscr{H} \to \{1, \ldots, K\}$ is a decision process that associates an arm b_t to a given history h_{t-1}:

$$b_t = \pi(h_{t-1}) \ .$$

The cumulated regret of a policy π at time t (after t pulls) is defined as follows:

$$R_t = t\mu^* - \sum_{t'=0}^{t-1} r_{t'} \ ,$$

where $\mu^* = \max_{k \in \{1,\ldots,K\}} \mu_k$ refers to the expected reward of the optimal arm to play at any time.

The objective is to find a policy that minimizes the expected cumulated regret, given as follows:

$$\mathbb{E}[R_t] = \sum_{k=1}^{K} (\mu^* - \mu_k)\, \mathbb{E}[T_k(t)], \tag{5.2}$$

where $\mu^* - \mu_k$ is the expected loss of playing arm k, and $T_k(t)$ refers to the number of times the arm k has been played from instant 0 to instant $t-1$.

A common approach for designing policies consists in assigning a numerical value, called *index*, to each arm based on its history h_t^k gathered at time t. The approach is the following:

- during the first K iterations, the arms $1, \ldots, K$ are sequentially played;
- at each subsequent time step t, for every machine $k \in \{1, \ldots, K\}$, an index is computed based on its history h_t^k;
- the arm with the largest index is selected to be pulled at time t.

One of the most famous such "index-based" policies is UCB1, proposed in [41].

5.6.2 Analysis of the similarities and differences with our problem

5.6.2.1 Similarities

Our problem of on-line selection of iterative sampling algorithms can be seen in some ways as a multi-armed bandit problem. Indeed, the set of different iterative sampling algorithms available in this context is the analog of the set of reward distributions available in the bandit context. An iterative sampling algorithm can be assimilated to a probability distribution since it identifies each dangerous contingency with a specific probability (defined intrinsically by the chosen parametrization of the algorithm). We also face the problem of selecting at one instant which is the best distribution given a history of observations. The approach used for solving our problem is also similar to an index-based bandit algorithm since, after having first called each of the n iterative sampling algorithms available, we compute at each step a score for each algorithm (the discovery factor) in order to select which one to call.

5.6.2.2 Differences

If we just wanted to pick up a maximum number of dangerous contingencies, the problem tackled in this chapter would actually be a bandit problem. However, by having as goal to pick a maximum number of *different* dangerous contingencies, it departs away from the standard multi-armed bandit problem. Indeed, it is as if we were dealing with a "new type" of multi-armed bandit problem where, once an arm has produced an event that was associated with a good reward, the reward associated to this event in future plays would be equal to zero, whatever the arm generating this event.

5.6.2.3 Discussion

Viewed in the light of these comparisons, the problem addressed in this chapter can be seen as a practical motivation for defining a new type of multi-armed bandit problem, which can be solved by the looping and discovery rate-based strategies we have developed. Moreover, it is likely that some of the results in this field could be used in future work to better understand the properties of the two strategies that have been proposed in this chapter or even to design more efficient ones.

Note that power system security assessment problems closed to the one considered here have already inspired a new sequential decision making problem in [43]. In particular, this latter paper proposes an algorithm for solving this problem based on the optimistic paradigm and the Good-Turing missing mass estimator. This algorithm has been shown to uniformly attain the optimal discovery rate in a macroscopic limit sense by using proof techniques developed for multi-armed bandit problems.

6

Estimating the probability and cardinality of the set of dangerous contingencies

We consider here the same context as in the rest of this thesis, i.e. large scale power system security assessment problems where the available computational resources are bounded and do not allow an exhaustive screening of the contingency space. We specifically focus on the case where this space is discrete, which is representative of most power system security assessment problems.
This chapter presents how the cross-entropy method for rare-event simulation can be used in this context for estimating the probability of the set of dangerous contingencies and how it is possible to use it for estimating the number of dangerous contingencies in a discrete search space.

Let X be a random variable taking its values in an event space $\mathscr{X}$ with a probability density function $f(\cdot)$, let $S(\cdot)$ be a real-valued function defined on $\mathscr{X}$ and γ be a real number. In the rare-event simulation context, one needs to estimate the probability of occurrence l of an event $\{S(X) \geq \gamma\}$, i.e. to estimate the expression $E_{X\sim f(\cdot)}\left[I_{\{S(X)\geq\gamma\}}\right]$.[1]

The problem of estimating the probability of occurrence of a dangerous contingency is exactly a rare-event problem where the event space $\mathscr{X}$ is our contingency space, the function S is our objective function O and the number γ plays the exact same role as the threshold γ defining a contingency as dangerous if $O(x) \geq \gamma$.

We will describe in this chapter how it is possible to estimate the probability of occurrence of a rare-event. In particular, we will first present importance sampling methods for rare-event simulation. Afterwards, we will introduce the cross-entropy method for rare-event simulation and provide a fully specified algorithm for estimating the probability of occurrence of an event $\{S(X) \geq \gamma\}$. We will then use this algorithm to adapt the basic iterative sampling algorithm proposed in this thesis and estimate the probability of occurrence of a dangerous contingency when the available computational resources are bounded. We will also explain how to use this estimate to compute the number of dangerous contingencies when considering problems where the contingency space is discrete, and provide illustrative simulation results.

6.1 Estimating the probability of occurrence of a rare-event

The material of this first section is largely borrowed from [23], to which we refer the reader for a complement of information.

6.1.1 Importance sampling for rare-event simulation

In rare-event simulation problems, the probability of occurrence l of an event $\{S(X) \geq \gamma\}$ is extremely low, say smaller than 10^{-6}, and estimating it with enough accuracy by relying on a Crude Monte Carlo (CMC) estimator

$$\hat{l} = \frac{1}{N}\sum_{j=1}^{N} I_{\{S(X_j)\geq\gamma\}} \tag{6.1}$$

[1] The function $I_{\{logical\_expression\}}$ is defined by $I_{\{logical\_expression\}} = 1$ if $logical\_expression = true$ and 0 otherwise. If $\mathscr{X}$ is finite, the expression $E_{X\sim f(\cdot)}\left[I_{\{S(X)\geq\gamma\}}\right]$ can be written equivalently as $\sum_{X\in\mathscr{X}} I_{\{S(X)\geq\gamma\}} f(X)$.

requires to draw a considerably large sample $X_1, X_2, \ldots, X_N$ from $f(\cdot)$. For example, estimating l, with a sample of size N leads to a standard error $\sigma_{\hat{l}} = \sqrt{\frac{l(1-l)}{N}}$. Hence, a sample size of $N \simeq 10^{10}$ is required in order to estimate $l \simeq 10^{-6}$ with relative error of 1% (i.e. with a standard error of $0.01 \cdot l$).

An alternative to CMC is based on importance sampling. With such an approach, a random sample $X_1, X_2, \ldots, X_N$ is drawn from an importance sampling distribution $g(\cdot)$ and the probability of occurrence of the event is estimated via the following estimator[2]

$$\hat{l} = \frac{1}{N} \sum_{j=1}^{N} I_{\{S(X_j) \geq \gamma\}} \frac{f(X_j)}{g(X_j)} \,. \tag{6.2}$$

In this context, the most effective way to estimate l would be to adopt the "ideal" importance sampling distribution

$$g^*(X) = \frac{I_{\{S(X) \geq \gamma\}} f(X)}{l} \,. \tag{6.3}$$

Indeed, since l is constant, using this "ideal" importance sampling distribution $g^*(\cdot)$ (6.3) would lead to an estimator (6.2) having a zero variance. Consequently, we would need to produce only a one element sample to determine l. The obvious difficulty is that $g^*(\cdot)$ depends on the unknown parameter l.

6.1.2 The cross-entropy method for rare-event simulation

The main idea of the cross-entropy (CE) method for rare event simulation is to find inside an a priori given set $\mathscr{G}$ of probability distributions defined on $\mathscr{X}$, an element $g(\cdot)$ such that its distance to the "ideal" sampling distribution is minimal. A convenient measure of distance between two probability distributions $a(\cdot)$ and $b(\cdot)$ on $\mathscr{X}$ is the Kullback-Leibler divergence, which is also named the cross-entropy between $a(\cdot)$ and $b(\cdot)$. The Kullback-Leibler divergence, which is not a distance in the formal sense since it is for example not symmetric, is defined as follows:

$$\mathscr{D}(a,b) = E_{X \sim a(\cdot)} \left[\ln \frac{a(X)}{b(X)} \right] \,. \tag{6.4}$$

The CE method reduces the problem of finding an appropriate importance sampling probability distribution to the following optimization problem:

$$\arg\min_{g \in \mathscr{G}} \mathscr{D}(g^*, g) \,. \tag{6.5}$$

[2] assuming that $g(X) \neq 0$ whenever $I_{\{S(X) \geq \gamma\}} f(X) \neq 0$

One can show through simple mathematical derivations that solving (6.5) is equivalent to solve

$$\arg\max_{g\in\mathscr{G}} E_{X\sim f(\cdot)}\left[I_{\{S(X)\geq\gamma\}}\ln g(X)\right]\ , \tag{6.6}$$

which does not depend explicitly on l anymore.

If l is not too small, CE-based algorithms for rare-event simulations estimate a good solution of (6.6) by solving its stochastic counterpart

$$\arg\max_{g\in\mathscr{G}} \sum_{j=1}^{M} I_{\{S(X_j)\geq\gamma\}}\ln g(X_j)\ , \tag{6.7}$$

where the sample $X_1, X_2, \ldots, X_M$ is drawn according to $f(\cdot)$. When l is too small, say $l < 10^{-6}$, which is often the case in rare-event simulation, the value of M one has to adopt for having a "good" stochastic counterpart may be prohibitively high and some specific iterative techniques need to be adopted to solve (6.6). The use of these techniques is often equivalent to solving a sequence of rare event problems using the same probability distribution $f(\cdot)$ and function S but with increasing values of γ converging to the value of γ related to the original problem.

Under some specific assumptions on $\mathscr{X}$, $f(\cdot)$ and $\mathscr{G}$, it is possible to solve analytically the optimization problem (6.7). This property is often exploited in the CE context.

Let us now suppose that $\mathscr{X}$ is $\mathbb{R}^n$ and let us denote by $Gauss_{\mathbb{R}^n}(\cdot, v)$, where $v = [\mu, \sigma] \in \mathbb{R}^n \times \mathbb{R}^n$, the n-dimensional (diagonal) Gaussian probability distribution

$$Gauss_{\mathbb{R}^n}(x, v) = \prod_{i=1}^{n} \frac{1}{\sigma[i]\sqrt{2\pi}} e^{-\frac{(x[i]-\mu[i])^2}{2\sigma[i]^2}}\ , \tag{6.8}$$

where $x[i]$ is the ith component of the random variable X and $\sigma[i]$ ($\mu[i]$) is the standard deviation (mean) of the n-dimensional probability distribution alongside the ith direction.

Then, one can show that if $f(\cdot)$ is a n-dimensional Gaussian probability distribution and $\mathscr{G}$ is the set of all n-dimensional Gaussian probability distributions, the solution

$Gauss_{\mathbb{R}^n}(\cdot, v^*)$ of (6.7) can be computed analytically:

$$\mu[i] = \frac{\sum_{j=1}^{M} I_{\{S(X_j)\geq\gamma\}} X_j[i]}{\sum_{j=1}^{M} I_{\{S(X_j)\geq\gamma\}}}, \tag{6.9}$$

$$\sigma[i] = \sqrt{\frac{\sum_{j=1}^{M} I_{\{S(X_j)\geq\gamma\}} (X_j[i] - \mu[i])^2}{\sum_{j=1}^{M} I_{\{S(X_j)\geq\gamma\}}}}. \tag{6.10}$$

6.1.3 An iterative CE-based rare-event simulation algorithm

As mentioned in the previous subsection, when l is too small, one needs to draw a prohibitively high number of samples to obtain a "good" stochastic counterpart (6.7) of (6.6). This originates from the fact that, in order to have a "good" stochastic counterpart, a sufficient number of samples X_j for which $S(X_j) \geq \gamma$ needs to be drawn. Iterative algorithms are therefore used for solving accurately this stochastic counterpart.

The rationale behind these algorithms is the following. First, let us observe that even if the probability distribution used to draw the sample used for building the stochastic counterpart was not drawn according to $f(\cdot)$ but well another probability distribution, called $h(\cdot)$, the stochastic counterpart could still have the same meaning provided that every term of the sum is weighted by a factor $\frac{f(X_j)}{h(X_j)}$. Therefore, if one can identify a probability distribution $h(X_j)$ for which the probability of occurrence of the event $\{S(X) \geq \gamma\}$ is not too small, it is very likely that the number of samples that will have to be drawn to obtain a good stochastic counterpart will have to be relatively small.

To identify such a distribution, it is of common practice to solve a sequence of rare-event problems differing only by the values of γ used. The first iteration of this sequence consists of a rare-event problem based on a small enough value of γ [3] so as to require only drawing a reasonable number of samples with f for having a "good" stochastic counterpart[4]. The probability distribution computed by solving this stochastic counterpart is in general more likely than f to generate events associated with high

[3] The smaller the value of γ is, the higher the probability of the event $\{S(X_j) \geq \gamma\}$ is.

[4] Actually, it is not required to use f to draw the sample at the first iteration, provided that the stochastic counterpart is corrected appropriately.

values of $S(\cdot)$. It is then used to generate the sample for solving the stochastic counterpart of the second rare-event problem, which differs only from the first one by a larger value of γ. This larger value of γ is itself defined from this same distribution to guarantee that by proceeding like this, not too many samples have to be drawn to solve accurately the stochastic counterpart at the second iteration. The algorithm proceeds similarly over the next iterations and stops when the rare-event problem solved is identical to the original one.

6.1.4 A fully specified algorithm for estimating the probability of occurrence of a rare-event

Figure 6.1 gives the tabular version of a fully specified CE-based algorithm for estimating the probability of occurrence of an event $\{S(X) \geq \gamma\}$ when $\mathscr{X}$ is a bounded subset of $\mathbb{R}^n$.

This algorithm is based on the iterative scheme described in previous subsection. It is particularized to the case where $\mathscr{G}$, the set of probability distributions in which one looks for an element which "stands" the closest to the "ideal" sampling distribution, is the set of Gaussian distributions "truncated" to values belonging to $\mathscr{X}$. Similarly to the notation adopted for denoting non-truncated Gaussian distributions, the symbol $Gauss_{\mathscr{X}}(\cdot, v)$ is chosen to refer to truncated ones. The value of these truncated distributions at $x \in \mathscr{X}$ is $\dfrac{Gauss_{\mathbb{R}^n}(x,v)}{E_{X \sim Gauss_{\mathbb{R}^n}(\cdot,v)}[I_{\{X \in \mathscr{X}\}}]} \cdot I_{\{X \in \mathscr{X}\}}$.

At every iteration t, the algorithm proceeds as follows. First, it draws from the distribution $Gauss_{\mathscr{X}}(\cdot, v_t)$ computed at the previous iteration a sample named $\mathscr{X}_t$. From this sample, it computes a value γ_t which is such that only a small fraction ρ (ρ is a parameter of the algorithm) of the elements of $x \in X_t$ lead to a value $S(x)$ larger or equal to γ_t. If the value of γ_t so computed turns out to be larger than γ, then it is replaced by γ. This value γ_t is then used together with $Gauss_{\mathscr{X}}(\cdot, v_t)$ to define the rare-event problem to be solved at iteration t. By defining the rare-event problem in this way, it is likely that the probability of the event $\{S(X) \geq \gamma_t\}$ with $X \sim Gauss_{\mathscr{X}}(\cdot, v_t)$ is not too small. The stochastic counterpart of the optimization problem (see Equations (6.7) and (6.6)) can therefore be defined by using a sample which is not too large. The algorithm described in Figure 6.1 uses the already drawn sample $\mathscr{X}_t$ to build this stochastic counterpart (where every term is weighted by $\dfrac{f(x)}{Gauss_{\mathscr{X}}(x, v_t)}$). An analytical solution, which is an approximation of the solution of the stochastic counterpart, is then used to compute the parameter $v_{t+1} = [\mu_{t+1}, \sigma_{t+1}]$ of the sampling distribution used at the next iteration.

The algorithm stops when γ_t is equal to γ. Before stopping, the algorithm com-

putes an estimate of l by exploiting the importance sampling estimator (6.2) with the importance sampling distribution $Gauss_{\mathscr{X}}(\cdot, v_{t+1})$. It returns both this estimate of l and $Gauss_{\mathscr{X}}(\cdot, v_{t+1})$, which usually gives strong preference to events x such that $S(x) \geq \gamma$.

Problem definition: an event space $\mathscr{X}$, a function $S : \mathscr{X} \to \mathbb{R}$ with $\mathscr{X} \subset \mathbb{R}^n$, a random variable $X \in \mathscr{X}$ taking its values in $\mathscr{X}$ with the distribution $f(\cdot)$ and a real number γ.

Algorithm parameters: $v_1 = [\mu_1, \sigma_1]$, C, ρ
Output: an estimation of the (small) probability $E_{X\sim f(\cdot)}\left[I_{\{S(X)\geq\gamma\}}\right]$ and a distribution $Gauss_{\mathscr{X}}(\cdot, v)$ giving preference to events x such that $S(x) \geq \gamma$.
Algorithm:

Step 1. Set t equal to 1. Set $nbElite$ equal to the largest integer inferior or equal to $\rho \times C \times n$. If $nbElite < 1$ then set $nbElite$ to 1.
Step 2. Set $\mathscr{X}_t$ equal to the empty set and r_t to an empty vector.
Step 3. Draw independently $N = C \times n$ elements according to the probability distribution $g_t(\cdot) = Gauss_{\mathscr{X}}(\cdot, v_t)$ and store them in $\mathscr{X}_t$.
Step 4. For every element $x \in \mathscr{X}_t$, compute $S(x)$ and add this value at the end of the vector r_t.
Step 5. Sort the vector r_t in decreasing order and set $\hat{\gamma}_t = \min(\gamma, r_t[nbElite])$.
Step 6. For $i = 1, 2, \ldots, N$, set $\mu_{t+1}[i] = \frac{\sum_{x\in\mathscr{X}_t} I_{\{S(x)\geq\hat{\gamma}_t\}} x[i] f(x)/g_t(x)}{\sum_{x\in\mathscr{X}_t} I_{\{S(x)\geq\hat{\gamma}_t\}} f(x)/g_t(x)}$ and

$\sigma_{t+1} = \sqrt{\frac{\sum_{x\in\mathscr{X}_t} I_{\{S(x)\geq\hat{\gamma}_t\}} (x[i]-\mu_{t+1}[i])^2 f(x)/g_t(x)}{\sum_{x\in\mathscr{X}_t} I_{\{S(x)\geq\hat{\gamma}_t\}} f(x)/g_t(x)}}$. Set $v_{t+1} = [\mu_{t+1}, \sigma_{t+1}]$.

Step 7. If $\hat{\gamma}_t = \gamma$, estimate l using the estimator
$\hat{l} = \frac{1}{N} \sum_{j=1}^{N} \frac{f(X_j)}{Gauss_{\mathscr{X}}(X_j, v_{t+1})} \cdot I_{\{S(X_j)\geq\gamma\}}$, where the samples X_j are drawn from $Gauss_{\mathscr{X}}(\cdot, v_{t+1})$, and return both $\hat{l}$ and the distribution $Gauss_{\mathscr{X}}(\cdot, v_{t+1})$. Else, set $t \leftarrow t + 1$ and go to **Step 2.**

Figure 6.1: A fully specified CE-based algorithm for estimating the probability of occurrence of an event $\{S(X) \geq \gamma\}$ when the event space is a bounded subset of $\mathbb{R}^n$.

Let us now elaborate on the role of the parameters of this CE algorithm:

- the parameter C determines the size of the samples $\mathscr{X}_t$ in a way that $|\mathscr{X}_t| = C \times n$. The rationale behind adopting a sample $\mathscr{X}_t$ whose cardinality is proportional to the dimension n of the event space $\mathscr{X}$ is that usually, the larger the dimension of event space space is, the larger the sample $\mathscr{X}_t$ has to be for the algorithm to behave well. A default value for this parameter C equal to 10 is usually adopted.

- as explained before, the parameter ρ determines the percentage of elements x of $\mathscr{X}_t$ for which $\{S(x) \geq \gamma_t\}$. A relevant default value for ρ is 0.1.

- finally, the choice of the Gaussian family of sampling distributions is essentially guided by practical considerations, namely the fact that it is easy to draw samples from such distributions and the fact that it leads to closed-form solutions of the minimization problem (6.7).

6.2 Estimating the probability of the set of dangerous contingencies

We now go back to the problem of identifying dangerous contingencies within a discrete contingency space and with limited computational resources on which we focus in this chapter. For the sake of simplicity, we assume that the contingencies are uniformly distributed in $\mathscr{X}$ (with probability $p = \frac{1}{|\mathscr{X}|}$).

Figure 6.2 proposes an adapted version of our basic iterative sampling algorithm (described in Figure 3.7), inspired from the CE-based algorithm for rare-event simulation introduced in Figure 6.1, for estimating the probability of the set of dangerous contingencies. This algorithm uses the last sampling distribution computed by the basic iterative sampling algorithm (whose parameters are denoted by λ_{final}) to generate a new sample of s points from the Euclidean embedding space, and then estimates the probability l with an estimator derived from the one used in the rare-event simulation context. In this estimator, whose formula is provided in Equation (6.11), each term corresponding to a dangerous contingency is weighted by the ratio between the probability p of the contingency $preImage(y)$ in $\mathscr{X}$ and the probability that this contingency is drawn by the final sampling distribution generated by the algorithm. This latter probability is equal to $\int_{z \in V_y} Gauss_{\mathbb{R}^n}(z, \lambda_{final}) dz$, where V_y is the Voronoi cell of the projection of $PreImage(y)$ in $\mathscr{Y}$.

$$\hat{l} = \frac{1}{s} \sum_{y \in S_{final}} I_{\{O(preImage(y)) \geq \gamma\}} \frac{p}{\int_{z \in V_y} Gauss_{\mathbb{R}^n}(z, \lambda_{final}) dz} \tag{6.11}$$

Problem definition: a discrete contingency space $\mathscr{X}$, a pre-image function $PreImage : \mathbb{R}^n \to \mathscr{X}$, an objective function $O : \mathscr{X} \to \mathbb{R}$ and a threshold $\gamma \in \mathbb{R}$.

Algorithm parameters: the parameters $\lambda_0 = [\mu_0, \Sigma_0]$ of the initial n-dimensional Gaussian sampling distribution, the size s of the sample drawn at each iteration and the number m of best solutions chosen at each iteration.
Input: the amount $res_{available}$ of available computational resources.
Output: an estimate $\hat{l}$ of the probability of the event $\{O(x) \geq \gamma\}$ (which corresponds to the probability that a contingency is dangerous).
Algorithm:

Step 1. Set $res = res_{available}$.
Step 2. Set P and $\mathscr{X}_{dang}$ to empty sets.

Step 3. Call $(\mathscr{X}_{dang}, P, res) = BIS(\mathscr{X}_{dang}, P, res)$ (see Figure 3.7).

Step 4. Using the last sampling distribution generated by the basic iterative sampling algorithm *BIS*, generate a new sample S_{final} of s points of $\mathbb{R}^n$ and estimate l with the estimator
$\hat{l} = \frac{1}{s} \sum_{y \in S_{final}} I_{\{O(PreImage(y)) \geq \gamma\}} \frac{p}{\int_{z \in V_y} Gauss_{\mathbb{R}^n}(z, \lambda_{final}) dz}$ where $p = \frac{1}{|\mathscr{X}|}$,
V_y is the (n-dimensional) Voronoi cell of the projection of the contingency $PreImage(y)$ in $\mathscr{Y}$ and λ_{final} are the parameters of the final sampling distribution used by the basic iterative sampling algorithm.
Step 4. Output $\hat{l}$ and stop.

Figure 6.2: Adaptation of the basic iterative sampling algorithm provided in Figure 3.7 for estimating the probability of occurrence of the set of dangerous contingencies when these latter belong to a finite discrete space, in which they are uniformly distributed.

6.3 Estimating the cardinality of the set of dangerous contingencies

We assume that the discrete contingency space considered in this chapter is finite. Subsequently, the set of dangerous contingencies is finite too, and its cardinality can be estimated in a straightforward way once the probability of the set of dangerous contingencies has been computed by the algorithm provided in the previous section. As we assume that the contingencies of $\mathscr{X}$ follow a uniform distribution, they all have the same probability of occurrence p (which is equal to the inverse of the cardinality of $\mathscr{X}$). By also assuming that these contingencies are mutually exclusive, the probability of the set of dangerous contingencies may be computed as:

$$l = \sum_{i=1}^{n_{dang}} p = n_{dang} \cdot p \,, \tag{6.12}$$

where n_{dang} is the total number of dangerous contingencies. This latter number can thus be estimated from the value of $\hat{l}$ as follows:

$$\hat{n}_{dang} = \frac{\hat{l}}{p} \,. \tag{6.13}$$

6.3.1 Illustration

We have implemented the algorithm given in Figure 6.2 on the case study that was treated in Chapter 4.3 (an $N-2$ analysis on the Belgian transmission system). The available computational budget has been chosen purposely high ($res_{available} = 2000$) so that the algorithm stops by itself, when all the points of the current sample are associated to the same contingency by the pre-image function.

We have performed 100 runs of this algorithm. For each of them, we have derived the value of $\hat{n}_{dang}$ from the value of $\hat{l}$ outputted by the algorithm according to Equation (6.13), and computed the average values of $\hat{l}$ and $\hat{n}_{dang}$ obtained over these 100 runs (that we denote by $\overline{\hat{l}}$ and $\overline{\hat{n}_{dang}}$), as well as the standard deviation of $\hat{n}_{dang}$. These results are reported in Table 6.1, where they are compared with those of a classical Monte Carlo sampling algorithm. The latter results were obtained by drawing 100 random sets of 749 contingencies (which corresponds to the average number of contingencies analyzed over the 100 runs of the iterative sampling algorithm when $\hat{l}$ is outputted). For each of them, we evaluated the value of $\hat{l}$ according to Equation (6.1) and subsequently the value of $\hat{n}_{dang}$ according to Equation (6.13). We finally computed the average values of $\hat{l}$ and $\hat{n}_{dang}$ estimated over these 100 runs.

Table 6.1: Estimation of the probability and cardinality of the set of dangerous contingencies with the iterative sampling algorithm provided in Figure 6.2 and with a classical Monte Carlo sampling algorithm, for an $N-2$ analysis of the Belgian transmission system.

	$\bar{\hat{l}}$	$\overline{\hat{n}_{dang}}$	$\sigma(\hat{n}_{dang})$
Iterative sampling	$1.03 \cdot 10^{-3}$	207.4	6.7
Monte Carlo sampling	$5.47 \cdot 10^{-6}$	1.1	2.5

We observe that our iterative sampling algorithm significantly outperforms the Monte Carlo sampling method, given that there are in practice 210 dangerous contingencies. These results illustrate the fact that a crude Monte Carlo estimator requires to draw a large sample from the contingency space so as to provide an accurate estimate of l and does not yield good results when the sample at hand is too small. To the contrary, our iterative sampling approach provides a very good estimation of the number of dangerous contingencies in the search space with the same computational budget.

7

Conclusion

7.1 Contributions of this thesis

The present dissertation gathers research contributions in the field of large scale power system security assessment.

The main contribution of this work is:

- a comprehensive iterative sampling framework for identifying dangerous contingencies within wide contingency spaces with bounded computational resources.

As sub-contributions linked to the development of this approach, we have also proposed:

- a procedure for embedding a discrete contingency space in a Euclidean space.
- a way to adapt this iterative sampling algorithm in order to estimate the probability and cardinality of a finite set of dangerous contingencies.
- several ways to define over the contingency space a real-valued function (the objective function) reflecting the severity of each contingency;

Finally, the case where several such iterative sampling algorithms are available and can be run sequentially has been considered. For this case, we have designed:

- strategies for selecting on-line which of the available iterative sampling algorithms to execute at each step so as to take better advantage of the available computational resources and identify as many dangerous contingencies as possible.

7.2 Further research directions

We propose hereafter several directions to enrich the work that has been done in this thesis.

7.2.1 Extension of the metrization procedure

It comes as immediate future work to extend the metrization process described in Chapter 3, especially for dealing with various types of contingencies within the same study (e.g., $N-k$ contingencies involving different values of k, or a mix of $N-k$ equipment outage contingencies and shifts in the generation and load patterns).

7.2.2 Development of performance guarantees

The different iterative sampling frameworks proposed in this thesis have been meant to identify as many dangerous contingencies as possible while minimizing the risk of missing some dangerous contingencies.

For instance, the algorithms provided in Chapters 2.4 and 3.4 propose to repeat the basic iterative sampling algorithm that was introduced in Chapter 2.3 several times – as long as the available computational resources have not been exhausted – which is a good way to globally reduce the variance of the performances of each run of this algorithm. Besides, we have suggested in Chapter 5 to split the search space into several sub-areas, define an iterative sampling algorithm initialized in each of these sub-areas and execute them sequentially in an order determined by the proposed selection strategies. This procedure seems to be a relevant solution to make sure that no subpart of the search space is undeservedly leaved unexplored, and thus avoid missing too many dangerous contingencies.

However, these approaches do not come with any performance guarantees. One way to obtain strict performance guarantees for this approach would be to try to transpose theoretical results obtained in the field of stochastic optimization or sequential decision making to the problem at hand.

7.2.3 Extension of the simulations to larger systems

The method provided in this thesis to address large scale security assessment problems has already been implemented on large real problems on which it has proven to efficiently identify dangerous contingencies when the available computational budget is limited. A straightforward extension of this work would be to apply it to even larger case studies, like the European interconnected transmission system. This could for instance allow to identify failure modes that would not have been suspected a priori, which has not been the case with the security assessment problems studied in this thesis since they have already been studied very thoroughly.

7.2.4 Integration into TSO's research environments

The following step in the development of the approach proposed in this thesis would be to integrate it into TSO's research environments, in such a way that it could be used in combination with the other security assessment tools at hand.

Appendix A

Pseudo-geographical representations of power system buses by multidimensional scaling

This appendix investigates new possibilities for visualizing power systems. The new representations proposed here were derived from our work when setting up the process of metrization of the contingency space (presented in Chapter 2).

The work presented in this appendix has been published in the Proceedings of the 15th International Conference on Intelligent System Applications to Power Systems (ISAP 2009) [44].

Graphical representations of power systems are systematically used for planning and operation. The coordinate systems commonly used by Transmission System Operators are static and often reflect the geographical position of each equipment of the system. We propose in this appendix to position on a two-dimensional map the different buses of a power system in a way such that their coordinates also highlight some other physical information related to them. These pseudo-geographical representations are computed by formulating multidimensional scaling problems which aim at mapping a distance matrix combining both geographical and physical information into a vector of two-dimensional bus coordinates. We illustrate through examples that these pseudo-geographical representations can help to gain insights into the power system physical properties.

A.1 Introduction

Two-dimensional representations of power systems based on the geographical location of their devices have always been used for power system planning and operation. Originally, the - now universally-used - one-line diagrams were created as simplified representations of three-phase power systems. The need to incorporate in these diagrams some indications about the physical properties of the system quickly arose. Solutions using line widths were proposed to represent the power flows in the transmission lines [45]. Color shadings were used to represent limit violations. Also the availability status of the equipments was visualized, thanks to full or dotted lines.

As a matter of fact, a considerable effort was made over the last decade to enhance power system visualizations by exploiting available, both hardware and software, computational resources [46]. For example, color contours are now used to represent voltage magnitude variations across wide areas. Flows in transmission lines are visualized on pie charts, showing their loading, and this information can be supplemented by animated arrows showing the direction of the flows. These indicators are resized to reflect the state of the system. In parallel, solutions have recently been developed to match the one-line diagram of the system and the geographical information about the equipments, based on satellite views [47].

However, all the currently used representations use static and mostly geographically based coordinate systems to locate the various devices of the power system in their two-dimensional geometry.

We propose in this work a new methodology in order to enrich the existing two-dimensional graphical representations of power systems with any kind of useful information about their physical properties, in such a way that the locations of the buses of the system not only represent their geographical relation but also the variation of these

physical properties. Our approach is based on using multidimensional scaling as a tool to transform composite distance measures combining both geographical and physical information about power system buses into two-dimensional coordinates. The physical information incorporated into these distance measurements can be diverse. As we will illustrate, they can for example represent the reduced impedance of the network between two buses, their electromechanical distance [48] or distances that quantify any kind of interesting physical information (like, e.g., nodal sensitivity factors) associated to these two buses.

Such "pseudo-geographical" representations can convey meanings, interpretations and knowledge on the power system, and therefore facilitate the planning and the operation of the system. As way of example, let us suppose that the distance between the buses on the map also represents the reduced impedance of the system between each pair of buses, and that the resulting map shows that the buses are located in two main zones which are rather distant from each other. This can be an indication that low frequency oscillations may occur between these two zones of the system or that specific devices should be installed to modulate the power transfer between these zones (e.g., Thyristor Controlled Series Capacitor).

To compute the pseudo-geographical coordinates, we rely technically on multidimensional scaling methods which are often used in information visualization for exploring similarities or dissimilarities in high-dimensional datasets [35]. We have adapted these methods to be able to control the trade-off between the fidelity of representing the geographical and the physical information.

The rest of this appendix is organized as follows. Section A.2 describes the optimization problem lying behind the construction of these pseudo-geographical representations of power system buses. Two concrete application cases are introduced in Section A.3. In Section A.4, an algorithm is proposed to solve the optimization problem described previously. Section A.5 presents the representations obtained for the two illustration cases and conclusions are drawn in Section A.6.

A.2 Problem statement

The first step in the construction of a pseudo-geographical representation of the buses of a power system is the choice of the data to represent. For this task, the reader can refer to the illustrations provided in Section A.3. The chosen data have to be expressed as distances between the buses of the system. If we consider a power system with n buses, the input data consist of a set of $\frac{n(n-1)}{2}$ distances, one distance being defined for each pair of buses. For the sake of simplicity, we suppose that these distances are

represented in a distance matrix format. This matrix, denoted by D, is a symmetric n-by-n matrix of real non-negative elements. An element d_{ij} of matrix D corresponds to the distance between buses i and j. The diagonal terms of this matrix are identically null since the distance between one bus and itself is equal to zero.

Given this matrix of pairwise distances, it is possible to compute a set of two-dimensional coordinates $\left\{(x_k, y_k)\right\}_{k=1}^{n}$ for the buses such that the Euclidean distances between these coordinates approximate the distances given in matrix D. This can be done by solving the following optimization problem:

$$\underset{(x_1,y_1),\ldots,(x_n,y_n)}{\arg\min} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - d_{ij} \right)^2 . \tag{A.1}$$

To the set of node coordinates which is solution of this optimization problem, corresponds a pseudo-geographical representation of the buses of the power system. There are several things to comment about the problem expressed in Equation (A.1). First, the objective function is always positive and, generally, even strictly positive. In the particular case where the matrix D gathers the geographical distances between buses, its minimum is however equal to zero. The solution of (A.1) is also non-unique. Indeed, any pseudo-geographical map obtained by translating or rotating the pseudo-geographical map computed when solving (A.1) is also solution of (A.1) since such transformations leave the inter-bus distances unchanged.

Because we are not really interested to have a pseudo-geographical map that represents the distances given in D but well the ratio between these distances, we consider that any pseudo-geographical map obtained by scaling a solution of (A.1) is also as good as a solution of (A.1). Among the (infinite size) set of pseudo-geographical maps defined by applying any combination of translation, rotation and scaling operators to a solution of (A.1), we have decided to select the one in which the pseudo-geographical coordinates of two particular buses – defined a priori and referred to as reference buses – coincide with their geographical coordinates. Deciding to have their position fixed in the pseudo-geographical representation provides the user with a key for interpreting the diagram.

Later in Section A.4.2, we will carefully describe the similarity transformation (which is a composition of a translation, a rotation and a homothety) to use to make the pseudo-geographical position of the two reference buses coincide with their geographical location.

Even with such a calibration, the resulting representation may be so different from the geographical one that the users may have difficulties to interpret the obtained map. In such cases, it would be more desirable to have a map corresponding to a suboptimal

solution of (A.1) but which looks more alike the geographical map. To address this problem, we propose to embed the geographical bus coordinates in the input data of problem (A.1). To do so, we suggest to replace the d_{ij} terms that appear in Equation (A.1) by terms expressing a "mixed" distance, denoted by d^m_{ij}, formed by taking a convex linear combination of the distance d_{ij} and the geographical distance between buses i and j. If we denote by d^{geo}_{ij} this geographical distance, the mixed distance can be formulated as:

$$d^m_{ij} = \lambda d_{ij} + (1-\lambda) d^{geo}_{ij} \,, \tag{A.2}$$

where the parameter $\lambda \in [0,1]$. The value assigned to this parameter reflects the importance that is given to the distances d_{ij} with respect to the distances d^{geo}_{ij}.

The optimization problem lying behind the computation of a representation based on these mixed distances writes:

$$\underset{(x_1,y_1),\ldots,(x_n,y_n)}{\arg\min} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(\sqrt{(x_i-x_j)^2+(y_i-y_j)^2} - \left(\lambda d_{ij} + (1-\lambda) d^{geo}_{ij} \right) \right)^2 . \tag{A.3}$$

If we set $\lambda = 0$, the coordinates of the buses will only depend on their geographical distances and, among all solutions of the argmin problem, there will be the set of geographical coordinates of the buses. To the contrary, if λ is set equal to 1, the inter-bus geographical distances will not be taken into account, and, as a consequence, the pseudo-geographical map might be difficult to interpret. When λ increases within $[0,1]$, the relative weight of the data contained in matrix D with respect to the geographical distances between buses in the computed representation also increases.

A.3 Examples of application cases

We give in this section two concrete examples of data on which we will illustrate our approach for creating pseudo-geographical representations.

A.3.1 Visualizing the reduced impedances between buses

The first set of data is the set of the reduced admittances between each pair of buses of the system. The reduced impedance between two buses is obtained by reducing the

admittance matrix of the network to these two buses, and by computing the modulus of the inverse of this value.

For a power system with n buses, this procedure yields $\frac{n(n-1)}{2}$ values (one for each pair of buses) from which a distance matrix D can be built. These reduced impedances can be seen as electrical distances between buses since they provide a good image of "how distant" two buses are on an electrical point of view.

The electrical distance between two buses reflects for instance how likely their voltage angles are close or how a short-circuit at a specific bus will affect the currents arriving at the other buses. Visualizing this information can certainly help to get better insights into the power system physics.

A.3.2 Visualizing the voltage sensitivities of the buses

For this second example, we choose to work with voltage variations from static security analyses. The variations are computed by considering always the same base case configuration and by running a power flow to compute the voltage drops induced by the loss of a generator. If we assume that the generators are numbered from 1 to n_g and if we denote by ΔV_i^g the variation of the voltage magnitude at bus i when generator g is lost, we can associate to each bus a vector $\Delta V_i = \left(\Delta V_i^g\right)_{g=1}^{n_g}$ collecting its voltage variations.

In order to create a pseudo-geographical representation of these voltage sensitivities, the information contained in vectors ΔV_i has to be converted into inter-bus distances. On this purpose, we compute the Euclidean distance between these "voltage variation vectors". The distance between buses i and j is thus set equal to : $d_{ij} = \sqrt{\sum_{g=1}^{n_g} \left(\Delta V_i^g - \Delta V_j^g\right)^2}$.

These distance measurements express the dissimilarities between the voltage variations at the different buses after a loss of generation.

A.4 Computational method

We describe in this section our approach for solving the optimization problem given by Equation (A.3) (see Section A.2). We first develop the algorithm used to compute a solution this problem. Note that this algorithm has already been provided in Chapter 3.3 for computing "electrical" bus coordinates. We explain it again in this appendix for the sake of clarity. Afterwards, we carefully detail the similarity transformation to apply to this solution in order to make the pseudo-geographical position of the two reference buses coincide with their geographical location.

A.4.1 Resolution of the optimization problem

The first stage of approach is based on an optimization algorithm borrowed from the multidimensional scaling (MDS) literature. This algorithm is known as the SMACOF algorithm [36] (the acronym SMACOF stands for "Scaling by Majorizing a Complicated Function"). It will be explained at the end of this section after having introduced the mathematical background on which it relies.

Let us denote by $X^{geo} \in \mathbb{R}^{n\times 2}$ the matrix of the geographical coordinates of the buses, by $D \in \mathbb{R}^{n\times n}$ the matrix containing the inter-bus distance measurements and by $X \in \mathbb{R}^{n\times 2}$ the set of pseudo-geographical coordinates we want to compute.

The problem (A.3) can equivalently be written as:

$$\arg\min_{X} f(X) \,, \tag{A.4}$$

where

$$f(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\sqrt{\sum_{k=1}^{2} (x_{ik} - x_{jk})^2} - d_{ij}^m \right)^2 . \tag{A.5}$$

The function f defined in Equation (A.5) has the form of the classical stress function one commonly seeks to minimize in an MDS problem. It can be expanded as follows:

$$\begin{aligned} f(X) = &\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=1}^{2} (x_{ik} - x_{jk})^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(d_{ij}^m\right)^2 \\ &-2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\sqrt{\sum_{k=1}^{2} (x_{ik} - x_{jk})^2} \right) d_{ij}^m \,. \end{aligned} \tag{A.6}$$

The first term of this sum can also be written:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=1}^{2} (x_{ik} - x_{jk})^2 = tr(X'AX) \,, \tag{A.7}$$

with $A \in \mathbb{R}^{n\times n}$ being such that $a_{ii} = n-1$ and $a_{ij} = a_{ji} = -1$.

The second term of $f(X)$ does not depend on X and can be seen as a constant, so we set:

$$k_0 = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(d_{ij}^m\right)^2 . \tag{A.8}$$

In the third term of $f(X)$, we denote by $dist_{i,j}(X)$ the Euclidean distance between buses i and j:

$$\sqrt{\sum_{k=1}^{2}(x_{ik}-x_{jk})^2} = dist_{i,j}(X)\ . \tag{A.9}$$

Given (A.7), (A.8) and (A.9), Equation (A.6) can be written concisely as:

$$f(X) = tr(X'AX) + k_0 - 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} dist_{i,j}(X)\, d_{ij}^m\ . \tag{A.10}$$

The third term of this expression is non-convex and makes the resolution of the problem (A.4) difficult. To address this problem, one can majorize this term by a convex expression to get a new objective function, easier to minimize. The SMACOF algorithm exploits the following majorization, based on the Cauchy-Schwartz inequality:

$$\begin{aligned}\sum_{k=1}^{2}(x_{ik}-x_{jk})(y_{ik}-y_{jk}) &\leq \Big(\sum_{k=1}^{2}(x_{ik}-x_{jk})^2\Big)^{1/2} \\ &\quad \times\Big(\sum_{k=1}^{2}(y_{ik}-y_{jk})^2\Big)^{1/2} \\ &\leq dist_{ij}(X)dist_{ij}(Y)\ ,\end{aligned} \tag{A.11}$$

where $Y \in \mathbb{R}^{n\times 2}$ can be interpreted as another set of coordinates for the buses.

If we multiply both sides of the inequality by (-1) and divide by $dist_{ij}(Y)$, we obtain:

$$-dist_{ij}(X) \leq \frac{\sum_{k=1}^{2}(x_{ik}-x_{jk})(y_{ik}-y_{jk})}{dist_{ij}(Y)}\ . \tag{A.12}$$

By summing over $i = 1\ldots n$ and $j = i+1\ldots n$ we obtain the majorizing expression:

$$\begin{aligned}-2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} dist_{i,j}(X)\, d_{ij}^m &\leq -2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=1}^{2}\frac{d_{ij}^m}{dist_{ij}(Y)} \\ &\quad \times(x_{ik}-x_{jk})(y_{ik}-y_{jk}) \\ &\leq -2tr(X'B(Y)Y)\ ,\end{aligned} \tag{A.13}$$

with $B(Y) \in \mathbb{R}^{n \times n}$ being such that:

$$b_{ij} = \begin{cases} -\dfrac{d_{ij}^m}{dist_{ij}(Y)} & \text{for } i \neq j \text{ and } dist_{ij}(Y) \neq 0 \\ 0 & \text{for } i \neq j \text{ and } dist_{ij}(Y) = 0 \end{cases}$$

$$b_{ii} = -\sum_{j=1, j \neq i}^{n} b_{ij} \, . \tag{A.14}$$

By combining (A.10) and (A.13), the function f itself can be majorized:

$$f(X) \leq tr(X'AX) + k_0 - 2tr(X'B(Y)Y) = g(X) \, . \tag{A.15}$$

The function g is a quadratic function of X. The minimum of the function g is obtained when its derivative is equal to zero, i.e.:

$$\nabla g(X) = 2AX - 2B(Y)Y = 0 \, . \tag{A.16}$$

The value of X minimizing $g(X)$ is such that:

$$AX = B(Y)Y \, . \tag{A.17}$$

As the inverse A^{-1} does not exist since A is not full rank, this linear equation in X cannot be solved by premultiplying both sides of (A.17) by A^{-1}. The Moore-Penrose inverse, given by $A^+ = (A + \mathbf{1}_{n,n})^{-1} - n^{-2}\mathbf{1}_{n,n}$ (where $\mathbf{1}_{n,n}$ is the matrix such that $\mathbf{1}_{n,n}(i,j) = 1 \; \forall (i,j) \in \{1, \ldots, n\}^2$), is used in the SMACOF algorithm. The matrix X minimizing $g(X)$, and subsequently $f(X)$, is the following:

$$X = A^+ B(Y)Y \, . \tag{A.18}$$

It can be shown that the solution computed from (A.18) is such that $f(X) \leq f(Y)$. The SMACOF algorithm exploits this property to iteratively compute solutions with decreasing values of f. The solution computed at iteration i, denoted by X_i, is equal to $A^+ B(X_{i-1})X_{i-1}$. The tabular version of the procedure used in our simulations is given in Figure A.1.

Problem definition: an n-by-n distance matrix, a matrix $X^{geo} \in \mathbb{R}^{n\times 2}$ of geographical coordinates.
Algorithm parameters: a small positive value ε, which is the minimum decrease of f after an iteration for not stopping the iterative process, and a maximal number of iterations $IterMax$.

Output: a matrix $X^{MDS} \in \mathbb{R}^{n\times 2}$ of pseudo-geographical coordinates.

Algorithm:

Step 1. Set $X_0 = X^{geo}$.
Set iteration counter $Iter = 0$.
Step 2. Compute $f_0 = f(X_0)$. Set $f_{-1} = f_0$.
Step 3. While $Iter = 0$ or $\big((f_{Iter-1} - f_{Iter}) > \varepsilon$ and $Iter \leq IterMax\big)$ do:
Set $Iter \leftarrow Iter + 1$.
Compute $B(X_{Iter-1})$ by using Equation (A.14).
Set $X_{Iter} = A^{+}B(X_{Iter-1})X_{Iter-1}$.
Compute $f_{Iter} = f(X_{Iter})$.
Set $X_{Iter+1} = X_{Iter}$.
Step 4. Set $X^{MDS} = X_{Iter}$. Output X^{MDS}.

Figure A.1: A tabular version of the SMACOF algorithm for solving problem (A.4).

A.4.2 Geometrical transformation

In our simulation results, the coordinates X^{MDS} computed by the SMACOF algorithm will not be used as such to plot the data. To these coordinates will be applied successively a translation, a rotation and a homothetic transformation to have two buses matching their geographical position (see Section A.2). These transformations are detailed on Figure A.2 and are further illustrated on an example in Figure A.3.

Problem definition: the coordinates $X^{MDS} \in \mathbb{R}^{n\times 2}$ computed by the algorithm given on Figure A.1, the positions of the two reference buses on the geographical map. These positions are denoted by B_{r1}^{geo} and B_{r2}^{geo}. B_{r1}^{MDS} and B_{r2}^{MDS} refer to their position defined by X^{MDS}.
Output: a set of coordinates X such that the reference buses are positioned as in the geographical map.
Algorithm:

Step 1. Translate X^{MDS} along vector v such that $v = \overrightarrow{B_{r1}^{MDS} B_{r1}^{geo}}$.

Step 2. Apply to the resulting map a rotation around B_{r1}^{geo} of angle $\theta = angle\,(\overrightarrow{B_{r1}^{MDS} B_{r2}^{MDS}}, \overrightarrow{B_{r1}^{geo} B_{r2}^{geo}})$.

Step 3. Apply to the resulting map a homothetic transformation of origin B_{r1}^{geo} and of dilatation factor $k = \dfrac{length(B_{r1}^{geo} B_{r2}^{geo})}{length(B_{r1}^{MDS} B_{r2}^{MDS})}$.

Figure A.2: A fully specified algorithm for transforming the coordinates outputted by the multidimensional scaling algorithm so that the position of the reference buses B_{r1}^{MDS} and B_{r2}^{MDS} coincide with their geographical location.

On this latter figure, the four red crosses correspond to the geographical location of these points, and the four blue circles to their MDS location as outputted by algorithm given in Figure A.1. We apply successively the translation, rotation and homothetic operators defined on Figure A.2 to the blue circles, in order to make the blue circles 1 and 3 coincide with the 1 and 3 red crosses.

Figure A.3 shows the initial representations as well as those obtained after every stage of the similarity transformation.

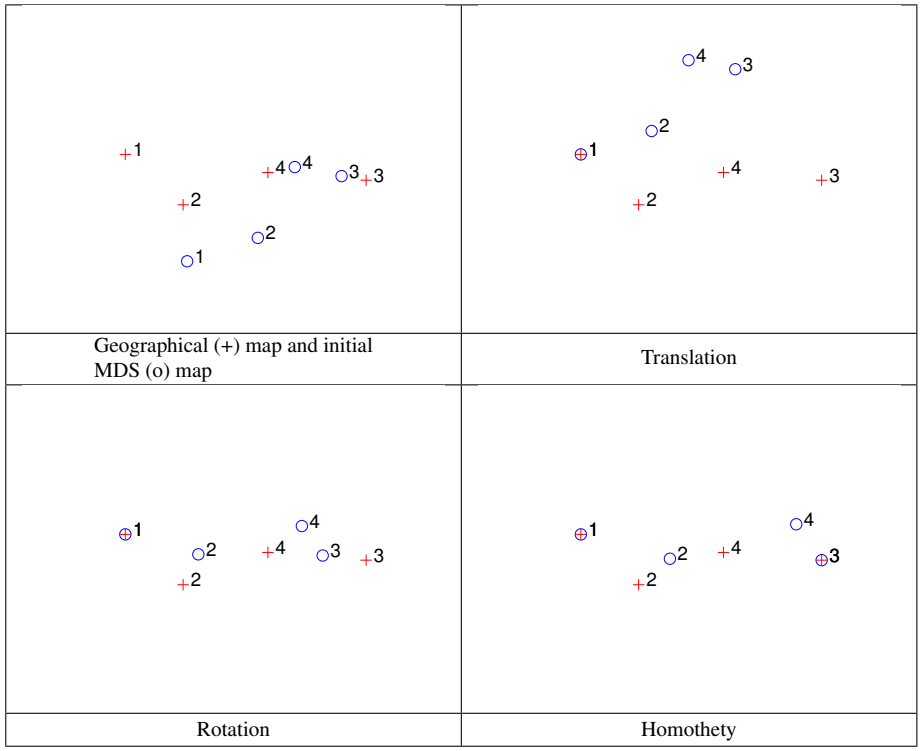


Figure A.3: Illustration of the similarity transformation (successive translation, rotation and homothety) for generating, from a set of coordinates computed by the SMACOF algorithm, a set of pseudo-geographical coordinates such that the points 1 and 3 of the pseudo-geographical map are positioned as in the geographical map.

A.5 Illustrations

We illustrate in this section our approach for computing pseudo-geographical representations of the two application cases introduced in Section A.3. The benchmark power system considered in this section is the IEEE 14 bus system [49], which has been vastly used in the literature as a test problem. Its one-line diagram is shown in Figure A.4. It is composed of five synchronous machines: two generation units are located at buses 1 and 2 respectively, and the three other machines, connected to buses 3, 6 and 8, are

synchronous compensators used only for reactive power supply. The 20 transmission lines of the system are either at 132 kV or 33 kV, the 33 kV part of the network being located at the top of Figure 4 and the transformers at buses 4, 5 and 7. The total amount of load is about 259 MW.

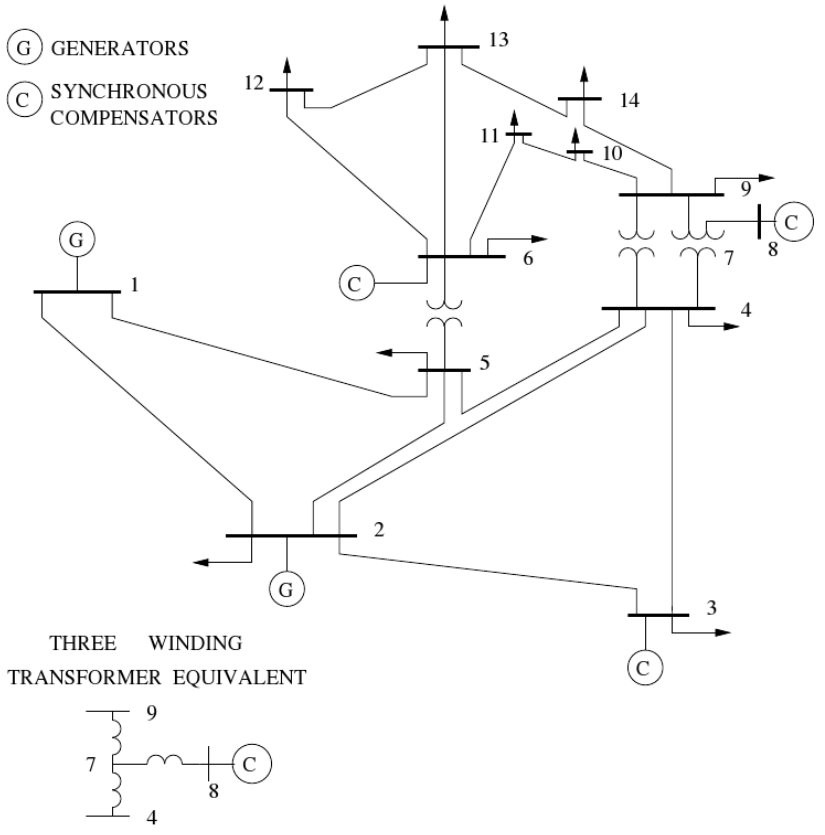


Figure A.4: IEEE 14 bus system.

A.5.1 Pseudo-geographical representation of the reduced impedances between buses

Figure A.5 reports the pseudo-geographical representations of the reduced impedances between the buses of IEEE 14 bus test system, for different values of the parameter λ. The case $\lambda = 0$ corresponds to the classical geographical representation of this network. The case $\lambda = 0.95$ corresponds to the representation of the buses according to a mixed distance as defined in (A.2). The case $\lambda = 1$ is a representation exclusively based on the electrical distances between buses.

Obviously, the geographical positions of the buses do not reflect their electrical distances. For example, buses 1, 2, 4 and 5 are much closer electrically than they are geographically. It is also worth noticing that bus 8, which is connected to buses 4 and 9 through a three windings transformer, appears quite close to these buses on the geographical representation while it is not the case on the pseudo-geographical ones. This was expected since the windings of a transformer generally have a rather high reactance whose value is in the range of the reactance value of a few tens of kilometers long transmission line. Similarly, we observe that the buses in the upper part of the system, which correspond to the lower voltage level (33 kV), are more remotely located when taking into account their electrical distances.

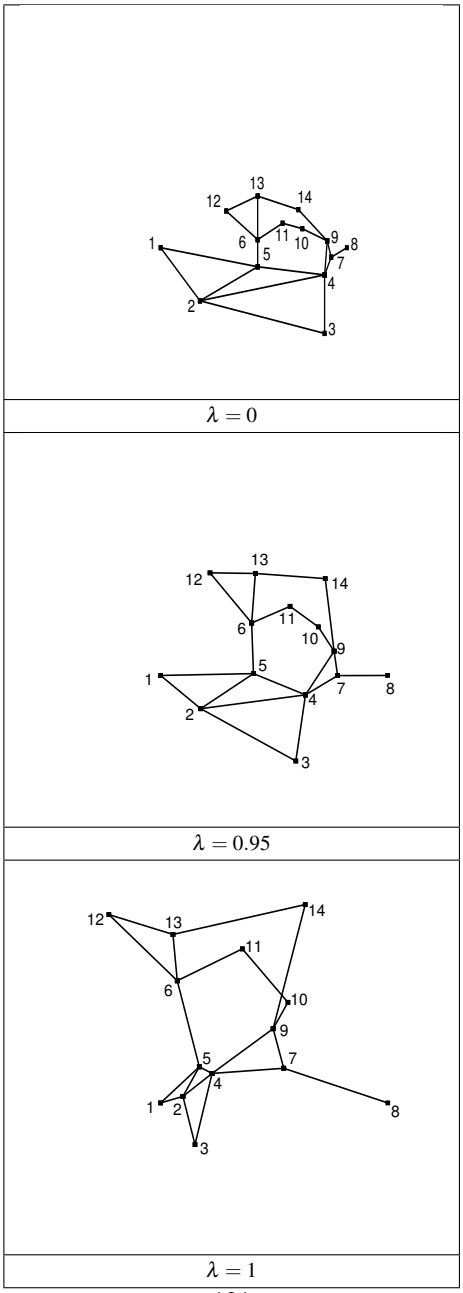


Figure A.5: Pseudo-geographical representations of the IEEE 14 bus system based on the reduced impedances. Buses 1 and 8 are taken as reference buses.

A.5.2 Pseudo-geographical representation of the voltage sensitivities of the buses

Figure A.6 illustrates the results of our experiments when considering the voltage sensitivities of the buses of IEEE 14 bus test system. As in the previous example, the three subfigures correspond to three increasing values of parameter λ (0, 0.8 and 1 respectively).

These figures clearly show that for large values of λ, the pseudo-geographical representations considerably differ from the pure geographical one.

We notice that the system has five synchronous machines controlling the voltage, which are connected to buses 1, 2, 3, 6 and 8. Since these buses are controlled in voltage except when considering the loss of the generator which is connected to them, their sensitivities are quite small and similar. It is therefore not surprising to see that they all appear next to each other on the map when $\lambda = 1$. Another interesting observation about this map is the far remote location of bus 4 with respect to the other elements of the system. This shows that the voltage variations caused by the loss of generation at this bus are different from the ones at the other buses. Notice that, by plotting the voltage profiles, we have observed that whatever the generator lost, the voltage at bus 4 was the lowest of the bus voltages. Conversely, the bus 13 is located on the far other end of the representation; it turns out that this bus, which is located in the low voltage part of the system, experiences also strong voltage variations but which are de-correlated to those of bus 4.

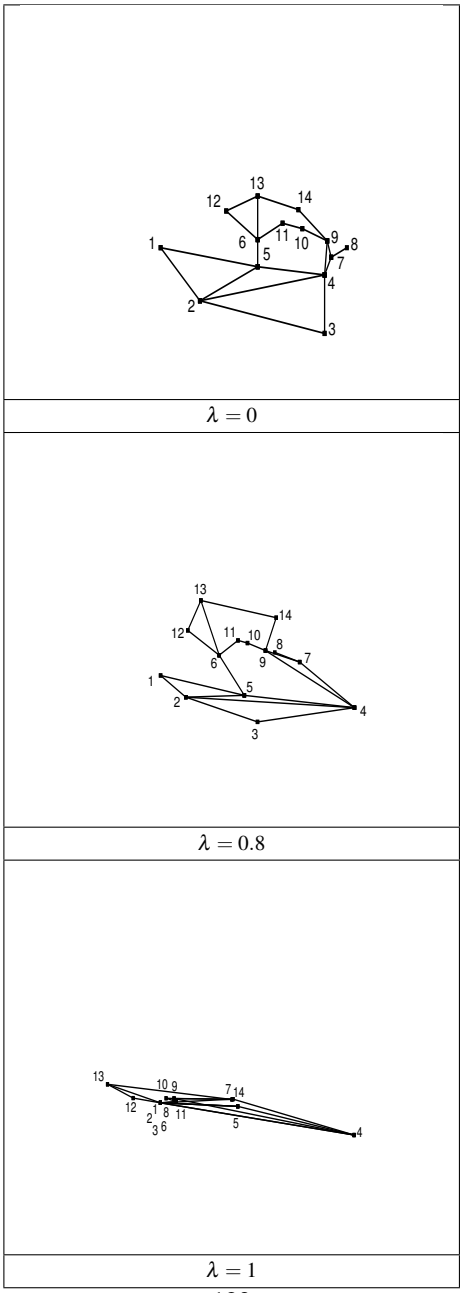


Figure A.6: Pseudo-geographical representations of the IEEE 14 bus system based on the voltage sensitivities. Buses 1 and 4 are taken as reference buses.

A.6 Conclusion

We have proposed in this appendix a new approach for visualizing power system data that can be expressed as pseudo-distances between buses of the system. The approach works by processing these data with a multi-dimensional scaling algorithm to obtain a two-dimensional map where the different buses are positioned in a way that the distance between every two buses represents these pseudo-distances. The approach has been illustrated on two case studies demonstrating the interest of mixing geographical and physical information to design power system representations. The results have shown that these pseudo-geographical representations can complement existing visualization tools for planning and operation of a power system.

Bibliography

[1] T.E. Dy Liacco. The adaptive reliability control system. *IEEE Trans. PAS*, PAS-86(5):517–531, May 1967.

[2] L.H. Fink and K. Carlsen. Operating under stress and strain. *IEEE Spectrum*, 15(3):48–53, March 1978.

[3] M. Pavella and P.G. Murthy. *Transient Stability of Power Systems: Theory and Practice*. John Wiley & Sons, 1994.

[4] T. van Cutsem and C. Vournas. *Voltage Stability of Electric Power Systems*. Technology and Engineering. Springer, 1998.

[5] P. Kundur. *Power System Stability and Control*. McGraw-Hill, 1994.

[6] J.J. Grainger and W.D. Stevenson. *Power System Analysis*. McGraw-Hill, New York, 1994.

[7] J.D. Glover and M.S. Sarma. *Power System Analysis and Design*. Brooks/Cole Publishing Co., 2001.

[8] U.G. Knight. *Power System in Emergencies*. John Wiley & Sons, 2001.

[9] D. Ernst, D. Ruiz-Vega, M. Pavella, P. Hirsch, and D. Sobajic. A unified approach to transient stability contingency filtering, ranking and assessment. *IEEE Transactions on Power Systems*, 16(3):435–444, 2001.

[10] Y. Zhang, L. Wehenkel, P. Rousseaux, and M. Pavella. SIME: A hybrid approach to fast transient stability assessment and contingency selection. *International Journal of Electrical Power and Energy Systems*, 19(3):195–208, 1997.

[11] M. Pavella, D. Ernst, and D. Ruiz-Vega. *Transient Stability of Power Systems. A Unified Approach to Assessment and Control*. Power Electronics and Power Systems. Kluwer Academic Publishers, 2000.

[12] D.P. Nedic. *Simulation of Large System Disturbances*. PhD thesis, UMIST, 2003.

[13] D.S. Kirschen and D.P. Nedic. Consideration of hidden failures in security analysis. In *Proceedings of the Power System Computation Conference*, Sevilla, 2002.

[14] Q. Chen. *The Probability, Identification and Prevention of Rare-Events in Power Systems*. PhD thesis, Iowa State University, 2004.

[15] Q. Chen and J.D. McCalley. Identifying high-risk N-k contingencies for on-line security assessment. *IEEE Transactions on Power Systems*, 20(2):823–834, 2005.

[16] W.A. Thompson. *Point Process Models with Applications to Safety and Reliability*. Chapman and Hall, 1988.

[17] V. Donde, V. Lopez, B. Lesieutre, A. Pinar, C. Yang, and J. Meza. Identification of severe multiple contingencies in electric power networks. In *Proceedings of the North American Power Symposium*, 2005.

[18] A.R. Conn, K. Scheinberg, and L.N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM series on optimization. Society for Industrial and Applied Mathematics, Mathematical Programming Society, 2009.

[19] F. Glover and G.A. Kochenberger. *Handbook of Metaheuristics*. International series in operations research & management science. Kluwer Academic Publisher, 2003.

[20] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[21] H.-G Beyer. *The Theory of Evolution Strategies*. Natural Computing Series. Springer, 2001.

[22] H. Muehlenbein, T. Mahnig, and A. Ochoa Rodriguez. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5:215–247, 1999.

[23] R.Y. Rubinstein and D.P. Kroese. *The Cross-Entropy Method. A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. Information Science and Statistics. Springer, 2004.

[24] R. Bardenet and B. Kégl. Surrogating the surrogate: accelerating Gaussian-process-based global optimization with a mixture cross-entropy algorithm. In *Proceedings ofthe 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

[25] R. Billinton, J. Oteng-Adjei, and R. Ghajar. Comparison of two alternate methods to establish an interrupted energy assessment rate. *IEEE Transactions on Power Systems*, 2(3):751–757, 1987.

[26] S.A. Ali, G. Wacker, and R. Billinton. Determination and use of sector and composite customer damage functions. In *1999 IEEE Canadian Conference on Electrical and Computer Engineering*, volume 3, pages 1483–1488, 1999.

[27] K.K. Kariuki and R.N. Allan. Evaluation of reliability worth and value of lost load. *IEEE Proceedings on Generation, Transmission and Distribution*, 143(2):171–180, 1996.

[28] J.C. Chow, R. Fischl, and H. Yan. On the evaluation of voltage collapse criteria. *IEEE Transaction on Power Systems*, 5:612–620, 1990.

[29] P. Kessel and H. Glavitsch. Estimating the voltage stability of a power system. *IEEE Transactions on Power System*, 1(3):346–354, 1986.

[30] P.-A. Lof, G. Andersson, and D.J. Hill. Voltage stability indices for stressed power systems. *IEEE Transactions on Power Systems*, 8:326–335, 1993.

[31] Y. Chen, C.W. Chang, and C.C. Liu. Efficient methods for identifying weak nodes in electrical power networks. *IEE Proceedings Generation, Transmission and Distribution*, 142:317–322, 1995.

[32] IEEE 30 bus test system: available at `http://www.ee.washington.edu/research/pstca/pf30/pg_tca30bus.htm`.

[33] F. Belmudes, D. Ernst, and L. Wehenkel. Cross-entropy based rare-event simulation for the identification of dangerous events in power systems. In *Proceedings of the 10th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS-08)*, Rincon, Puerto Rico, 2008. 8 pages.

[34] A.J. Wood and B.F. Wollenberg. *Power Generation, Operation and Control*. John Wiley & Sons, 1996.

[35] I. Borg and P. Groenen. *Modern Mutlidimensional Scaling: Theory and Applications*. Springer New York, 2005.

[36] J. de Leeuw. Applications of Convex Analysis to Multidimensional Scaling. In *Recent Developments in Statistics*, pages 133–146. North Holland Publishing Company, Amsterdam, 1977.

[37] F. Fonteneau-Belmudes, D. Ernst, and L. Wehenkel. A rare event approach to build security analysis tools when $N-k$ ($k>1$) analyses are needed (as they are in large scale power systems). In *Proceedings of the 2009 IEEE Bucharest PowerTech Conference*, Bucharest, Romania, 2009. 8 pages.

[38] F. Fonteneau-Belmudes, D. Ernst, C. Druet, P. Panciatici, and L. Wehenkel. Consequence driven decomposition of large scale power system security analysis. In *Proceedings of the 2010 IREP Symposium - Bulk Power Systems Dynamics and Control - VIII*, Buzios, Rio de Janeiro, Brazil, 2010. 8 pages.

[39] IEEE 118 bus test system: available at `http://www.ee.washington.edu/research/pstca/pf118/pg_tca118bus.htm`.

[40] T.L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[41] P. Auer, P. Fisher, and N. Cesa-Bianchi. Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, 47:273–280, 2002.

[42] R. Agrawal. Sample mean based index policies with O(log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, pages 1054–1078, 1995.

[43] S. Bubeck, D. Ernst, and A. Garivier. Optimal discovery with probabilistic expert advice. *Submitted*, arXiv:1110.5447.

[44] F. Fonteneau-Belmudes, D. Ernst, and L. Wehenkel. Pseudo-geographical representations of power system buses by multidimensional scaling. In *Proceedings of the 15th International Conference on Intelligent System Applications to Power Systems (ISAP 2009)*, Curitiba, Brazil, 2009. 6 pages.

[45] P.M. Mahadev and R.D. Christie. Envisioning power system data: concepts and a prototype system state representation. *IEEE Transactions on Power Systems*, 8(3):1084–90, August 1993.

[46] T.J. Overbye and D.A. Wiegmann. Reducing the risk of major blackouts through improved power system visualization. In *Proc. 2005 Power Systems Computational Conference (PSCC), Liège, Belgium, August 2005*.

[47] F.B. Lemos and F. Kober. Experience on development and utilization of Google™ Earth in a DMS environment. Abstract presented at the *10th International Workshop on Electric Power Control Centers (EPCC)*, 2009.

[48] R. Belhomme and M. Pavella. A composite electromechanical distance approach to transient stability. *IEEE Transactions on Power Systems*, 6:622–631, 1991.

[49] IEEE 14 bus test system: available at `http://www.ee.washington.edu/research/pstca/pf14/pg_tca14bus.htm`.